How can I ...

build a motor control application?

System Technical Guide
Motor Control Management

Develop
your project

Schneider Electric

Disclaimer

This document is not comprehensive for any systems using the given architecture and does not absolve users of their duty to uphold the safety requirements for the equipment used in their systems or compliance with both national or international safety laws and regulations.

Readers are considered to already know how to use the products described in this document.

This document does not replace any specific product documentation.

© 2010 Schneider Electric. All rights reserved.

The STG Collection

System Technical Guides (STG) are designed to help project engineers and Alliance System Integrators during the development of a project. The STGs support users during the architecture selection and the project execution (design, configuration, implementation and operation) phases, with an introduction to the system operating modes.

Each STG is a starter kit that provides users with:

- Technical documentation
- Application examples
- Object libraries

Each STG addresses one or several customer challenges within the proposed solution using the offer from Schneider Electric.

All explanations and applications have been developed by both Schneider Electric experts and system integrators in our solution labs. The contributions from the system integrators help the kit's content meet the expectations of our users.

All STGs are illustrated with industry-specific applications to give more concrete examples of the methodology.

The STGs are not intended to be used as substitutes for the technical documentation related to the individual components, but rather to complement these materials and training.

Development Environment

Each STG has been developed in one of our solution platform labs using a typical PlantStruxure architecture.

PlantStruxure, the process automation system from Schneider Electric, is a collaborative system that allows industrial and infrastructure companies to meet their automation needs while at the same time addressing their growing energy management requirements. In a single environment, measured energy and process data can be analyzed to yield a holistically optimized plant.

Table of Contents

1. Introduction 7

1.1. Purpose 7

1.2. Customer Challenges 7

1.3. Prerequisites 8

1.4. Project Methodology 8

1.5. Project Description 10

2. Selection Phase 13

2.1. Introduction 13

2.2. Selection Principle 13

2.3. Selecting Starter Mode 16

2.4. Selecting Motor Control Devices 24

2.5. Selecting Architecture 28

3. Design 33

3.1. Introduction 33

3.2. Operating Modes 33

3.3. Hardware Design 36

3.4. Software Design 45

4. Configuration 55

4.1. Introduction 55

4.2. PAC and Motor Control Device Configuration 56

4.3. SCADA System Configuration 89

4.4. ETG100 Configuration 92

5. Implementation 95

5.1. Hardware Implementation 95

5.2. DFB Implementation in the PAC 98

5.3. Unity Pro Program Structuring 112

5.4. SCADA Implementation 114

6. Operation 117

6.1. Introduction 117

6.2. Process Control and Diagnosis 117

6.3. Faulty Device Replacement (FDR) Operation 124

7. Performance 127

7.1. Application Response Time (ART) of Devices 127

7.2. Faulty Device Replacement (FDR) 128

1. Introduction

1.1. Purpose

The aim of this System Technical Guide (STG) is to provide recommendations, guidelines and examples to help you integrate and manage motor control applications within PlantStruxure architecture.

Each process control project has specific requirements and constraints that influence the development of motor control solutions.

1.2. Customer Challenges

For customers in industries that require motor control solutions, the challenges are to observe the:

- Project specifications: size of the plant, process complexity, level of monitoring, metering and diagnosis and so on.
- Functional constraints: the process characteristics impose constraints in terms of power, load types and power supplies…
- Operational constraints: plant productivity, process quality, design costs, operational costs…

This guide suggests best practices to address these challenges and highlights specific areas, including:

- Mixed solutions for motor control devices: The guide presents control solutions, from the simplest to the most advanced, to describe various monitoring and diagnostic capabilities.
- Communication consistency in the global system: The guide provides a solution that fits the requirements of the automation system (control and start, monitoring loads…) and management (consumption, operating time, preventive maintenance…).
- Sub-assembly modularity: The developments have been made with a re-use objective in order to facilitate process extensions and reduce design costs.

1.3. Prerequisites

We recommend users have knowledge of the following software:

- Unity Pro
- Vijeo Citect
- PowerSuite

1.4. Project Methodology

This STG explains the project methodology that includes the following phases: Selection, Design, Configuration, Implementation and Operation. This document provides a step-by-step methodology to create motor control solutions, based on PlantStruxure architecture. Here is an overview of this method:

- *Selection*

  This phase presents the various steps required to select the most appropriate starter components, as well as the automation architectures that perform the control.

  From the project specifications to the functional and operational constraints, the Selection phase helps to choose the starter mode, the motor control devices and finally, the whole architecture.

- *Design*

  This phase comprises three main parts:

  - Operating modes: Application operating modes are the most structuring elements in the automation system's Design phase. They define the application's hardware and software parts and act on all of the architectural components. Here, the objective is to propose operating modes that can address specific situations: starter management in a remote mode with a PAC or in local mode with buttons or panels.
  - Hardware: Specific wiring design is required depending on the previous operating modes. This part presents solutions notably for emergency stops and motor control devices
  - Software: Motor control applications require the design of objects located in different devices using various software. This section shows how to design applications in:
    - Vijeo Citect for SCADA
    - Unity Pro for PAC

- *Configuration*

  The aim of this phase is to provide key information for configuring the various system components (PAC, motor control devices, SCADA applications). The main purpose is to build a consistent system configuration with the description of all of the data exchanged between key components. This phase comprises the following parts:

  - Configuration of data to be exchanged periodically between PAC application and motor control devices.
  - Configuration of data and data types to be used both in PAC and SCADA applications.

- *Implementation*

  The main purpose of this phase is to explain how to implement the various components introduced in the Design phase:

  - PAC and motor control devices
  - Unity Pro program
  - SCADA

- *Operation*

  In this phase, available operations for process control and related motor control diagnoses are described.

- *Performance*

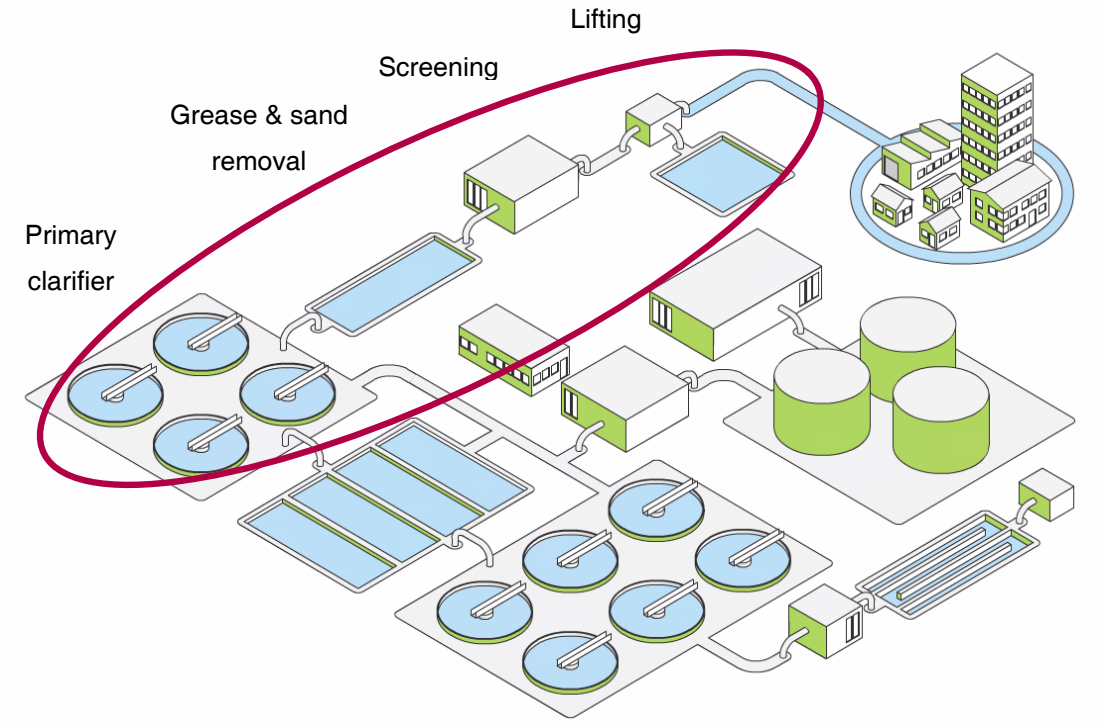
  This phase describes system performances:

  - Application Response Time (ART) of devices on an Ethernet network
  - Faulty Device Replacement (FDR), measurement of the time needed by the device to recover an operational state

1.5. Project Description

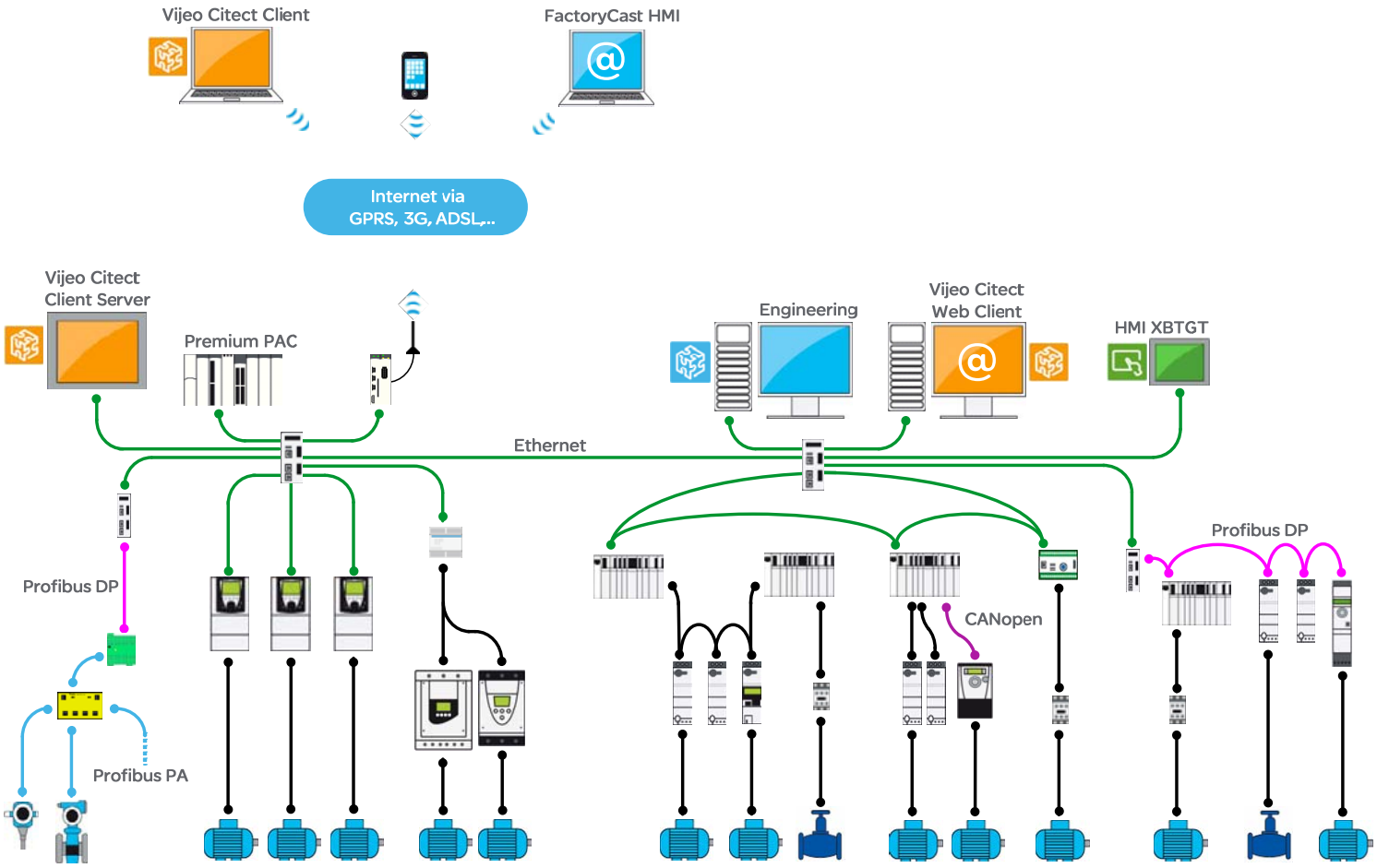
A process application has been developed to illustrate all explanations provided in this guide. It corresponds to the pre-treatment steps of a wastewater station, which is separated into 4 units:

- lifting
- screening
- grease & sand removal
- primary clarifier



The system architecture developed, presented in the diagram below, is based on a PlantStruxure control system, with a centralized Premium PAC and a Vijeo Citect SCADA system.

The complete description of the architecture is given in chapter 2.5.3 on Page 30.



2. Selection Phase

2.1. Introduction

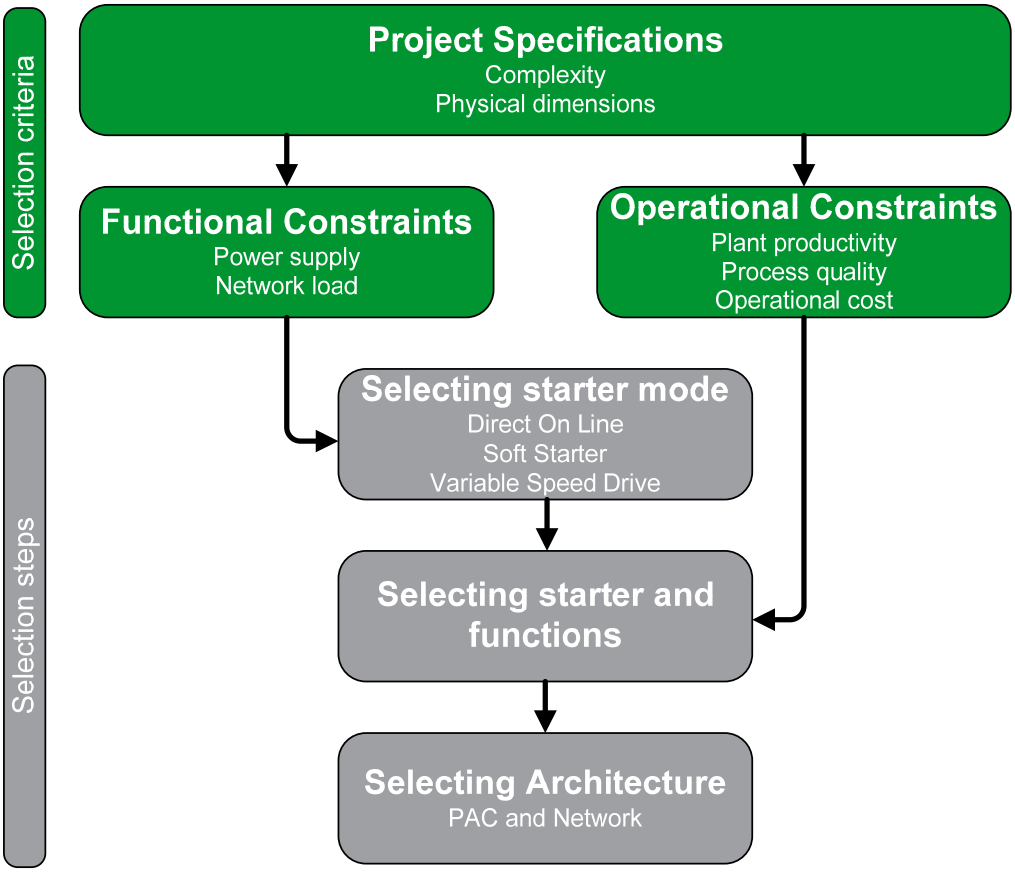
This chapter presents the various steps required to select the most appropriate starter components, as well as the automation architecture that performs the control.

2.2. Selection Principle

Each process control project has specific requirements and constraints that drive the selection of a motor control solution.

The project specifications present the characteristics that determine the selection criteria to be used in the selection steps.

The following diagram describes the selection procedure:



2.2.1. Project Specifications

The size of the plant, process complexity and other customer requirements define the PAC topology, I/Os architecture and connection types.

Project specifications also cover related constraints, which can be divided into two groups:

- Functional constraints

The process characteristics impose constraints in terms of power supply and network load that must be complied with.

These constraints are introduced in the following chapter without delving into the calculations or protection details of LV and MV networks.

Readers are advised to refer to the specific selection guides published by Schneider Electric for more details.

- Operational constraints

The project characteristics impose constraints such as:

  - Plant productivity: Traceability, environmental constraints
  - Process quality: Diagnostic information required, application standards
  - Operational cost: Operator profiles, energy monitoring

These functional and operational constraints are the inputs for the three following selection steps.

2.2.2. Selecting Starter Type

Three solutions are proposed to perform a motor control function:

- Direct On Line (DOL) starter:

This is the simplest solution, appropriate for low and medium power motors that do not require frequent start-up. Depending on requirements, elaborate protection and monitoring devices can be incorporated here.

- Soft starter:

For more powerful motors, the use of a soft starter allows reducing the load and torque of the motor during startup (and shut down, if necessary). This reduces the mechanical and electrical stress on the system.

- Variable Speed Drive (VSD):

This is the most powerful solution, allowing permanent control of the speed of a motor. It is used, for example, to regulate flows in pump applications. It can also help to optimize energy consumption.

The starter structure is detailed on Pages 17 to 23. A selection guide is provided on Page 24.

2.2.3. Selecting Motor Control Devices

The most appropriate device is selected based on the requested control functions. There are three types of functions:

- Motor protection: overload and short-circuit
- Metering functions: measurement of power, current and so on
- Monitoring functions: alarms, histories and so on

A table on Pages 26 and 27 summarizes the various starter functions.

2.2.4. Selecting Architecture

All selected motor control devices have to be connected to the global system architecture. Therefore, the communication link of the device must be selected in a consistent manner and in compliance with the chosen global architecture.

2.3. Selecting Starter Mode

2.3.1. Motor Starter Basic Functions

A motor starter unit has four basic functions:

- Isolation of the load from the main power supply
- Protection against short-circuit
- Protection against overload
- Control (start, stop, speed)

Each motor starter unit can be enhanced with additional functions depending on its system requirements:

- Power: speed controller, soft starter, phase reversal and so on
- Control: auxiliary contacts, time-delay, communication and so on

Starters are selected based on the power and control specifications.

2.3.2. Power Specifications Related to the Load

The choice of starter is determined by:

- Mechanical characteristics of the load (torque, inertia, speed)
- Power and electrical motor characteristics
- Necessary protections

These criteria are used to define one of the following solutions:

- Direct on line (DOL) starter
- Progressive start-up with soft starters
- Start-up at variable speed with variable speed drive (VSD)

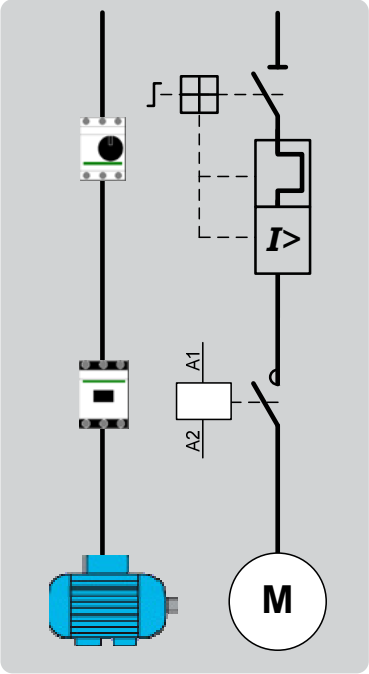
The step for selecting and dimensioning the power part of the starter is not developed in this document. Readers can refer to specific guides and Schneider Electric catalogs.

2.3.3. Direct On Line Starter

TeSys D with circuit breaker

This solution comprises a magneto-thermal breaker and a TeSys D that covers a power range of up to 110 kW. It provides the following basic functions:

- Protection against short circuit
- Protection against overload
- On/Off switch

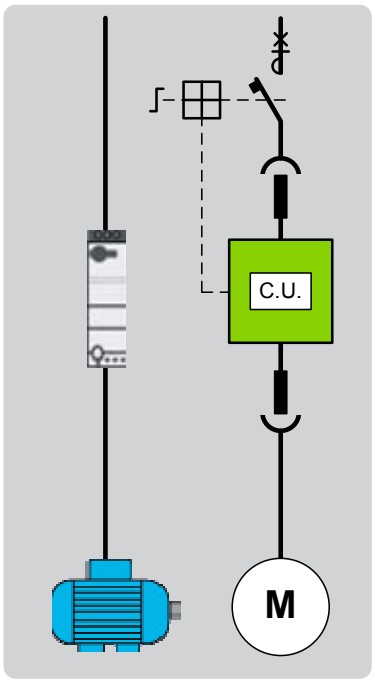


TeSys U Starter Controller

This is an integrated Direct On Line starter up to 15 kW, which performs the following functions:

- Protection and control of single-phase or 3-phase motors:
  - breaker function
  - overload and short-circuit protection
  - thermal overload protection and power switching
- Monitoring:
  - protection-function alarms
  - application monitoring (running time, number of detected faults, motor current values and so on)
  - logs (last 5 detected faults saved, with motor parameter values)

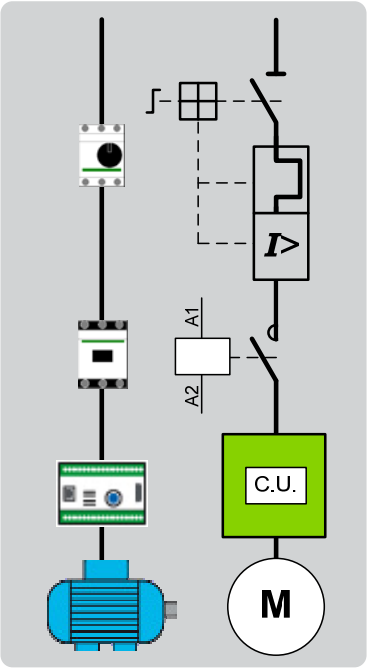
These functions can be added by selecting control units and function modules, which simply clip into the power base. Several communication modules are also available as an option and can be added to the power base. The product can therefore be customized at the last moment.



TeSys T Motor Management System

The capabilities of an over-current relay are limited when problems associated with voltage, temperature or special applications must be taken into account. The TeSys T provides complete management of the motor and its load. It incorporates these functions:

- Current and voltage sensors
- Hybrid analog and digital electronic technology
- Several communication buses available for data exchange and control
- Powerful motor modeling algorithms
- Application programs whose parameters can be set



2.3.4. Soft Starter

A soft start/soft stop unit is a controller that is used for the torque-controlled soft starting and stopping of three-phase squirrel-cage asynchronous motors. It offers soft starting and deceleration capabilities along with machine and motor protection and functions for communicating with control systems.

These functions are specially designed for use in applications such as pumps, fans and conveyors that are primarily used in Water and MMM industries.

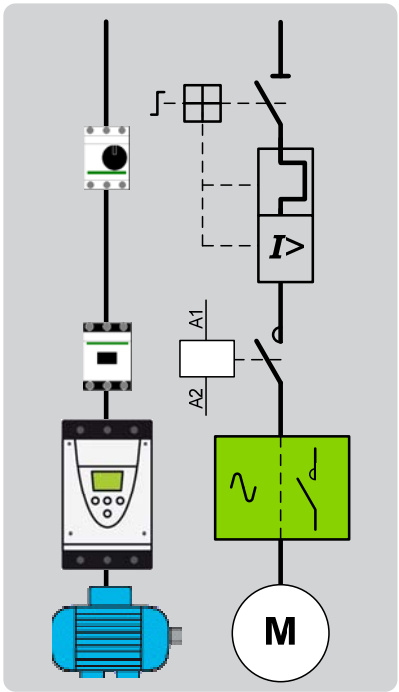
ATS 22 Soft Starter

This unit has a power range between 4 and 400kW. It is designed to operate in standard applications.

The bypass function (based on a bypass contactor) is more convenient to use thanks to its integration into the starter. The integrated contactor operates at the end of the startup sequence and deactivates once the stop command has been given to limit the starter's heat dissipation. This also leads to significant wiring reduction.

This unit also features thermal protection for motors as well as a monitoring facility for machines.

Note: Contactor and circuit breaker can be replaced by a TeSys U.

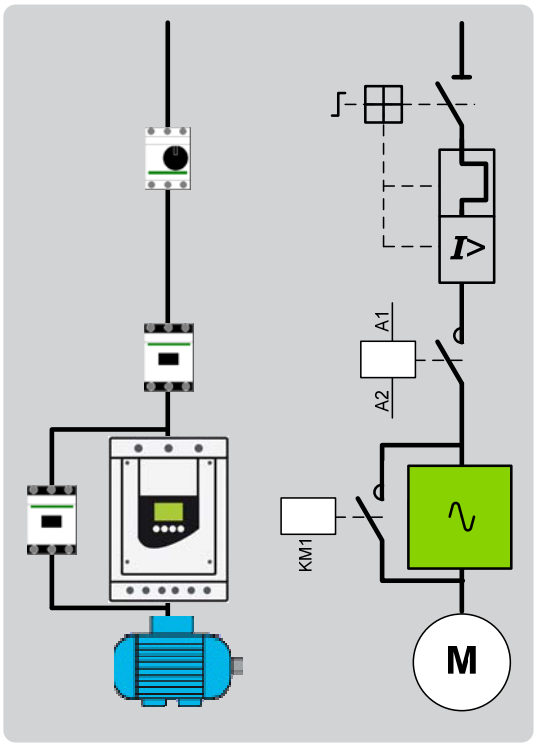


ATS 48 Soft Starter

This unit has a power range between 4 and 1200 kW. It is designed to operate in severe applications.

This unit also features thermal protection for motors as well as a monitoring facility for machines.

A contactor and circuit breaker can be replaced by a TeSys U controller associated with a specific module for a soft starter or variable speed drive.



Note: In the above figure, an external contactor is used to perform the bypass function of the ATS 48.

2.3.5. Variable Speed Drive Starter

ATV 312 Variable Speed Drive

The Altivar 312 drive is a frequency inverter for 200 to 600 V three-phase asynchronous motors from 0.18 to 15 kW. The Altivar 312 is robust, compact and easy to use. Its integrated functions are particularly suitable for the requirements of applications involving simple industrial machines. Modbus and CANopen protocols are integrated as standard.

ATV 32 Variable Speed Drive

The Altivar 32 drive is a frequency inverter for 200 to 500 V three-phase asynchronous and synchronous motors rated from 0.18 to 15 kW.

Up to 4 kW, the ATV 32 is available in a book format. In this power range, it can be installed side by side, saving a considerable amount of space.

Modbus and CANopen protocols are integrated into the ATV32 as standard. Additional communication cards handle MODBUS/TCP, PROFIBUS DP, DeviceNet and EtherCat protocols.

The Altivar 32 drive also offers the following functions:

- **Safety function** guarantees a high level of safety (IEC 61508 SIL 2).
- **Logic function** offers simple control system functions (Boolean, arithmetic operations, comparators and so on).
- **Bluetooth connectivity** - The SoMove Mobile software “transforms” any compatible mobile phone into a remote graphic display terminal by offering an identical Human-Machine Interface.

  SoMove Mobile software can be used to print out and save configurations, import them and export them to a PC or a hard drive equipped with Bluetooth.

ATV 61 Variable Speed Drive

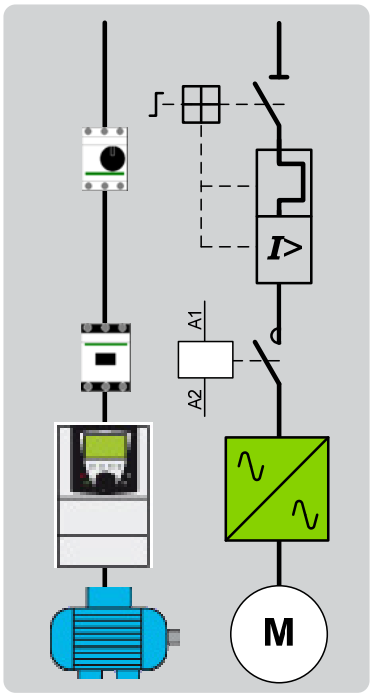
This is a frequency inverter for 3-phase asynchronous motors rated between 0.37 and 800 kW.

It has been designed for the most common fluid management applications in industrial and commercial buildings and infrastructures.

The Altivar 61 includes specific functions for pumping and ventilation applications:

- energy saving ratio, 2 or 5 point quadratic ratio
- automatic catching of a spinning load with speed detection
- adaptation of current limiting according to speed

- noise and resonance suppression due to the ability to adjust the switching frequency during operation
- preset speeds
- integrated PID regulator with preset PID references and automatic/manual ("Auto/Man") mode
- electricity and service hours meter
- fluid absence detection, zero flow and limited flow detection
- sleep function, wake-up function
- customer settings with display of physical measurements: bar, l/s, °C, and so on.



Note: The ATV 61 has been chosen here for its compatibility with pumping applications, but ATV 71 can also be used without any modification of the application.

2.4. Selecting Motor Control Devices

2.4.1. Selecting devices

The starter mode is closely linked to the load carried by the motor. The following table presents several typical examples in process control applications, such as water treatment or cement production.

Type of actuator	Description/ comment	Power range	Torque	Direct on line	Soft starter	VSD
Centrifugal pump	- Centrifugal pumps are used to cover a wide range of volume and pressure conditions.	1 kW to 10 kW	Quadratic	●	●	●
	- The flow can be controlled by using valves on the pump discharge manifold or by changing the rotation speed.	10 kW to 1MW			●	●
Dosing pump	- Dosing pumps are frequently used to inject fluids that may be difficult to mix efficiently in batch-tank system because of their low volume.	< 10 kW	Constant	●		●
Screw pump	- Screw pumps are also known as Archimede's screw. - They are used for lifting large volumes of fluid or material to a limited height. - They are driven through a speed reduction gear.	1 kW to 50 kW	Constant	●	●	
Mixer	- Mixers are used to give homogeneity to fluids. - Agitation is also used to speed up chemical process. - Mixing is performed by a propeller rotating in the fluid driven by a speed reduction gear.	1 kW to 50 kW	Constant		●	●
Moving devices	- Moving devices drive various types of mechanical systems such as: rotators, scrapers, shields, compressors, conveyors.	1 kW to 10 kW	Constant	●		●
Air blower and fan	- Air blowers or fans are used to provide air or oxygen for ventilation or aeration tank. - Flow can be adjusted using a mechanical system (fixed speed) or variable speed drive. Energy savings are possible by operating at reduced speed.	10 kW to 1 MW	Quadratic or constant		●	●
Mill and crusher	- Mills and crushers are used to grind materials. - They are typically high torque.	50 kW to 2 MW	Variable		●	●

2.4.2. Control Starter Functions

Depending on needs, it is necessary to be able to control some or all functions of a starter. The principal function groups are listed below:

- Motor Control Performance:

Control over power, torque, speed, reversing, start time and risk of jamming are required.

- Motor Protection:

Its purpose is to avoid operating motors in abnormal conditions that could result in negative events such as overheating, premature ageing, destruction of electrical windings, damage to coupling or gear box.

- Motor Metering and Monitoring Functions:

The purpose of implementing measurement devices is to provide continuous supervision of motor operating conditions. The collected data can be used effectively to improve energy efficiency and extend motor lifetime.

Monitoring functions allow you to control costs, schedule maintenance operations and keep historical information for legal requirements.

The following tables present a synthesis of the different device functions.

		DOL Starter			
		Motor circuit breaker + Contactor LC,D or F	Starter controller TeSys U		
			Standard control unit	Advanced control unit	Multifunction control unit
Motor Protection functions	Short circuit	●	●	●	●
	Overload	●		●	●
	Locked rotor	●	●	●	●
	No load running			●	●
	Earth fault			●	●
	Supply phases failure and imbalance	●	●	●	●
	Ventilation fault				
	Abnormal temperature rise				
	Shaft bearing seizure				
	Insulation fault				
	Long starting time	●	●	●	●
	Current phase reversal				
	Load Fluctuation (I, U, P)				
	Overtorque				
Metering functions	Indication of motor load			●	●
	Current on 3 phases (rms value)				●
	Average current				●
	Thermal capacity level				
	Motor temperature				
	Voltages on 3 phases				
	Frequency				
	Active power, power factor				
	Earth current				
	Motor torque				
Monitoring functions	Fault differentiation			●	●
	Remote or automatic thermal			●	●
	Local control, with I/O on product				●
	Local control, with HMI terminal				●
	Accelerating, decelerating torque control				
	Linear, S, U or customized acceleration and deceleration ramps				
	Bypass by contactor at starting end				
	Brake sequence				
	Automatic catching a spinning load, speed detection and automatic restart				
	Energy saving ratio, 2 point or 5 point quadratic ratio				
	Preset speed				
	Adaptation of current limiting according to speed				
	Noise and resonance suppression by switching frequency				
	Electricity and service hours meter				
	Detection of absence of fluid, detection of zero flow rate, limiting flow rate				
	Sleep function, wake up function				
	Customer settings with display of physical values: bar, l/s, °C, etc.				
	Safety function, integrated "power removal" SIL2				
	PI regulator and reference				
	Fault statistics: counters and history per type of protection				●
	Motor statistics: storage of motor statistics values				
	Diagnosis of faults affecting correct operation of the product				●
	Download and save configuration				

		DOL	Soft Starter		VSD		
		Motor management system TeSys T	ATS 48	ATS 22	ATV 312	ATV 61 ATV 71	ATV 32
Motor Protection functions	Short circuit	By Upstream Circuit Breaker					
	Overload	●	●	●	●	●	●
	Locked rotor	●	●		●	●	●
	No load running	●	●	●	●	●	●
	Earth fault	●		●	●	●	●
	Supply phases failure and imbalance	●	●	●	●	●	●
	Ventilation fault	●	●	●		●	
	Abnormal temperature rise	●	●	●		●	●
	Shaft bearing seizure	●	●			●	
	Insulation fault	●				●	●
	Long starting time	●	●	●	●	●	●
	Current phase reversal	●	●		●	●	●
	Load Fluctuation (I, U, P)	●	●			●	
	Overtorque		●	●	●	●	●
Metering functions	Indication of motor load	●	●		●	●	●
	Current on 3 phases (rms value)	●		●		●	●
	Average current	●	●			●	●
	Thermal capacity level	●	●	●	●	●	●
	Motor temperature	●				●	●
	Voltages on 3 phases	●					
	Frequency	●					
	Active power, power factor	●	●			●	
	Earth current	●				●	
	Motor torque		●			●	●
Monitoring functions	Fault differentiation	●	●	●	●	●	●
	Remote or automatic thermal	●	●	●	●	●	●
	Local control, with I/O on product	●	●	●	●	●	●
	Local control, with HMI terminal	●	●	●	●	●	●
	Accelerating, decelerating torque control		●	●	●	●	●
	Linear, S, U or customized acceleration and deceleration ramps				●	●	●
	Bypass by contactor at starting end		●	●			
	Brake sequence		●		●	●	●
	Automatic catching a spinning load, speed detection and automatic restart				●	●	●
	Energy saving ratio, 2 point or 5 point quadratic ratio					●	●
	Preset speed				●	●	●
	Adaptation of current limiting according to speed				●	●	●
	Noise and resonance suppression by switching frequency				●	●	●
	Electricity and service hours meter		●			●	●
	Detection of absence of fluid, detection of zero flow rate, limiting flow rate			●		●	
	Sleep function, wake up function					●	●
	Customer settings with display of physical values: bar, l/s, °C, etc.					●	
	Safety function, integrated "power removal" SIL2					●	●
	PI regulator and reference					●	●
	Fault statistics: counters and history per type of protection	●					
	Motor statistics: storage of motor statistics values	●				●	
	Diagnosis of faults affecting correct operation of the product	●				●	●
	Download and save configuration	●				●	●

2.5. Selecting Architecture

2.5.1. Introduction

The selection of starter mode and motor control device type was defined in the previous two steps. It is now time to build the final architecture that allows all motor control devices to be connected to the PAC and SCADA system.

As described above, the choice of architecture depends on the requested level of monitoring, metering and diagnostics and also depends on the compatibility required with the other parts of the system.

It has also been chosen to use the Faulty Device Replacement (FDR) service for this architecture in order to make the maintenance easier and more efficient.

Other selection criteria, such as cost and performance also influence the final choice.

2.5.2. Architecture Definition

We applied the following criteria to define the architecture example used to illustrate this guide. This architecture is detailed on Page 30, chapter 2.5.3.

- Mixed Solution of Motor Control Devices:

We have chosen several motor control solutions, from the simplest to the most advanced, in order to describe the various monitoring and diagnostics capabilities.

- Control and Management in the System:

A solution that fulfils the requirements of automation (motor control and starting, monitoring loads and so on) and management (consumption, operating time, preventive maintenance and so on) is provided.

- Sub Assembly Modularity:

Thanks to Advantys STB distributed IO islands modularity and their ability to be connected to different fieldbuses, we use these islands to communicate with several types of starters in order to facilitate process extension and reduce design cost.

- Evolution Facility:

The solution must be easily expandable beyond the initial design so that extensions can be integrated without having to reconsider the overall architecture.

The table below summarizes the different types of connections offered by the motor control devices:

		D.O.L starter					Soft Starter		Variable Speed Drive (VSD)		
		Motor circuit breaker + **Contactor** LC, D or F	Starter controller **TeSys U**			Motor management system **TeSys T**	**ATS 22**	**ATS 48**	**ATV 312**	**ATV 32**	**ATV 61 ATV 71**
			Standard	Advanced	Multifunction						
Type of links	Hard wired										
	Modbus SL										
	CANopen										
	Profibus DP										
	Ethernet Modbus TCP		*	*	*						
	EtherNet/IP										
	Advantys Pre wiring (EPI2145)										
	Advantys Internal bus										
	Advantys CANopen										

* On any TeSys U, Ethernet connectivity can be provided using an additional TeSysPort module.

- Faulty Device Replacement (FDR) service:

The FDR service allows automatic configuration of the replacement device's IP address and parameter settings.

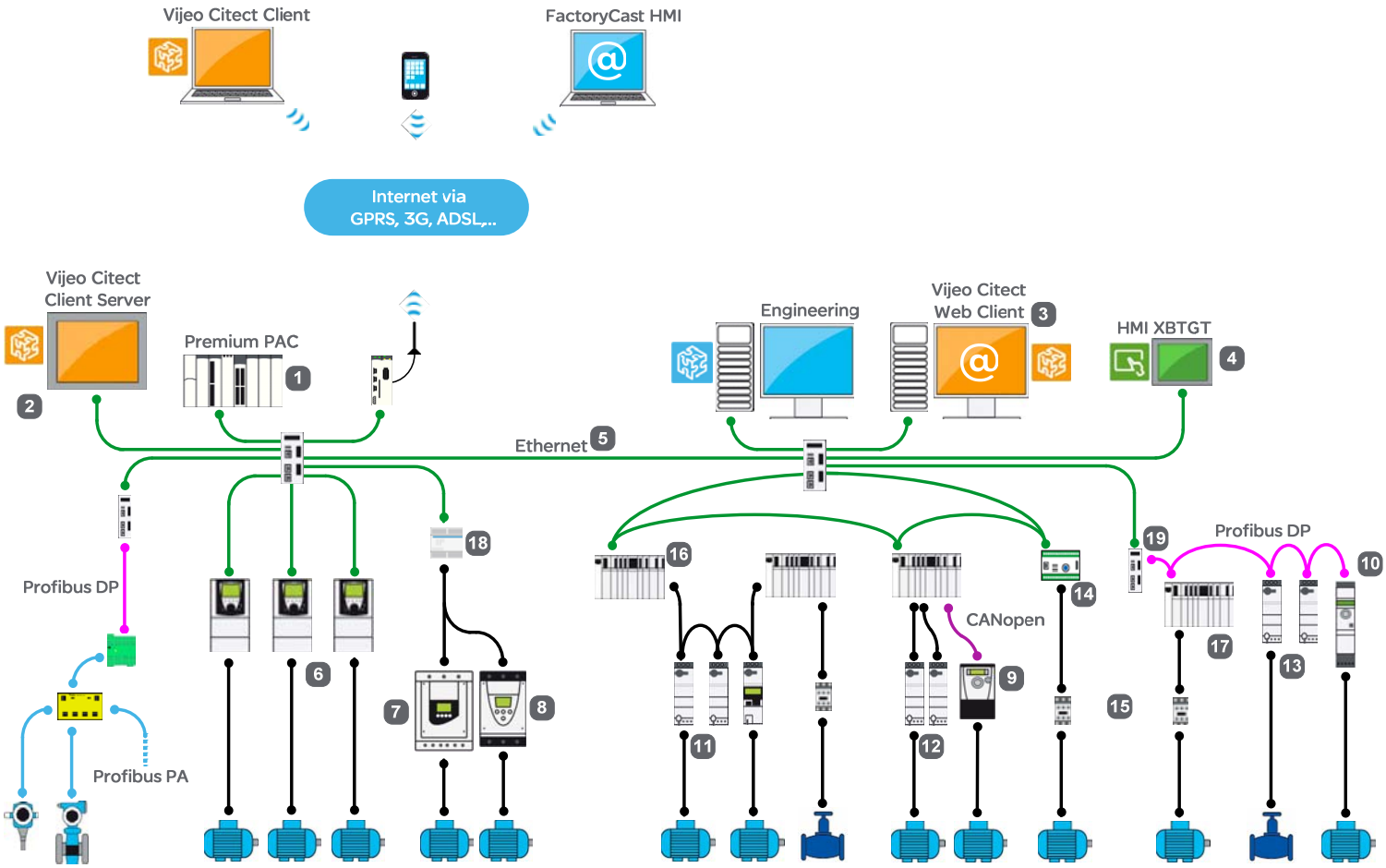
It makes the maintenance of devices connected on the Ethernet network easier.

The FDR service eliminates the need for service personnel to keep configuration records on hand and reduces the possibility of human error when entering a new configuration.

2.5.3. Architecture Example

The architecture example is based on a PlantStruxure control system, with a centralized Premium PAC (1), a standalone Vijeo Citect SCADA system (2) and a Vijeo Citect Web Client SCADA System (3). A Magelis HMI XBT-GT (4) is used for local control and monitoring (this STG does not describe the implementation of this equipment).

Note: Remote control rooms are not described in this STG.



An Ethernet network (5) connects all process steps in order to enable effective diagnostics and performance. Because of the Ethernet communication, the SCADA systems and HMI can easily access all process data. Embedded web diagnostic services, which are available from any standard web browser, facilitate the maintenance phase. The motor control devices are distributed across the plant with connections to the Ethernet network.

In the same way, Advantys STB islands (16) communicate with the controller via Ethernet and Profibus DP (17). These islands also connect DOL starters and other starters through various fieldbuses.

Advantys STB islands (16) and the TeSys T (14) controller are connected to an Ethernet daisy chain loop.

The following list describes the motor starters used in the architecture:

- ATV 61 (6): They are directly connected to the Ethernet network. All monitoring functions and controls are allowed. An embedded web server can be used for maintenance. Only the ATV 61 is presented in this guide but the ATV71 can also be used for the same purposes. The configuration and implementation steps are the same as for the ATV 61.
- ATS 48 (7): A Transparent Ready gateway ETG100 (18) is used to connect this device to the Ethernet network.
- ATS 22 (8): A Transparent Ready gateway ETG100 (18) is used to connect this device to the Ethernet network.
- ATV312 (9): The CANopen extension of an Advantys STB provides a cost-effective connection to ATV312 with a high level of diagnostics. Note that the CANopen port is embedded on ATV312.
- ATV32 (10): Profibus Remote Master (19) is used to connect this device to the Ethernet network.
- TeSys U with a Standard, Advanced and Multifunction control unit (11): In order to reach a high level of monitoring and metering, these TeSys U are connected to Advantys STB extension bus.
- TeSys U with a Standard control unit (12): A cost-effective solution is illustrated here with a pre-wired solution using the Advantys 2145 EPI module.
- TeSys U with a Standard and Multifunction control unit (13). These TeSys U are connected to a Profibus DP network and linked to the Ethernet network via a Profibus Remote Master (19).
- TeSys T (14): It is directly connected to the Ethernet network. All monitoring functions and controls are allowed. An embedded web server can be used for maintenance.
- TeSys D (15): The selected solution is wired directly to the Advantys STB I/O module or to the TeSys T (14).

This architecture combines various motor control device solutions that are detailed in the following chapters.

3. Design

3.1. Introduction

The aim of this chapter is to provide recommendations that facilitate the Design phase of your process control project.

It comprises three main parts:

- Description of global application operating modes
- Description of hardware design
- Description of software design

3.2. Operating Modes

3.2.1. Principle

Application operating modes are important structuring elements in the automation system definition phase. They act on all architecture components: SCADA, PAC and motor control device wiring.

Operating modes, described below, are rather general and can be easily adapted to the specific requirements of a project.

The modes defined in our project are summarized in the following table:

<table>
<tr><th>MODE</th><th colspan="2">Remote</th><th>Local</th><th>Off</th></tr>
<tr><td>SUB-MODE</td><td>Auto</td><td>Manual</td><td>-</td><td>-</td></tr>
<tr><td rowspan="2">ACTOR</td><td colspan="2">PAC</td><td rowspan="2">Local buttons</td><td rowspan="2">Local buttons</td></tr>
<tr><td>Application logic</td><td>SCADA/HMI</td></tr>
<tr><td>ACTION</td><td colspan="2">Run/Stop
Parameters modification</td><td>Run/Stop
Lights</td><td>Actuator powered off</td></tr>
</table>

3.2.2. Description

Remote Mode

In Remote mode, the command to the actuators is always sent from the PAC. Nevertheless, we differentiate two sub-modes:

- Auto mode: The PAC application logic directly controls the actuators.
- Manual mode: The operator controls the actuators via SCADA/HMI.

When switching from the Auto to Manual mode using SCADA/HMI, actuators maintain their states (they continue to run if they were running previously, and keep the same speed for motors controlled by variable speed drives).

When the system comprises many control stations (SCADA and HMI), it is necessary to define an access rights policy in order to avoid conflicts when operating in the Manual mode.

In some specific applications, Manual mode can be deactivated in order to suppress any human intervention on the process actuators.

Local Mode

In the Local mode, the command to the actuator is sent directly from local controls. Therefore, actuators can be controlled even in the case of an inoperative PAC. That means that local controls are hardwired directly to the pre-actuators (contactor, variable speed drive, and so on).

The commands are usually sent using switch buttons. Depending on the drives, a remote graphic display terminal can be used.

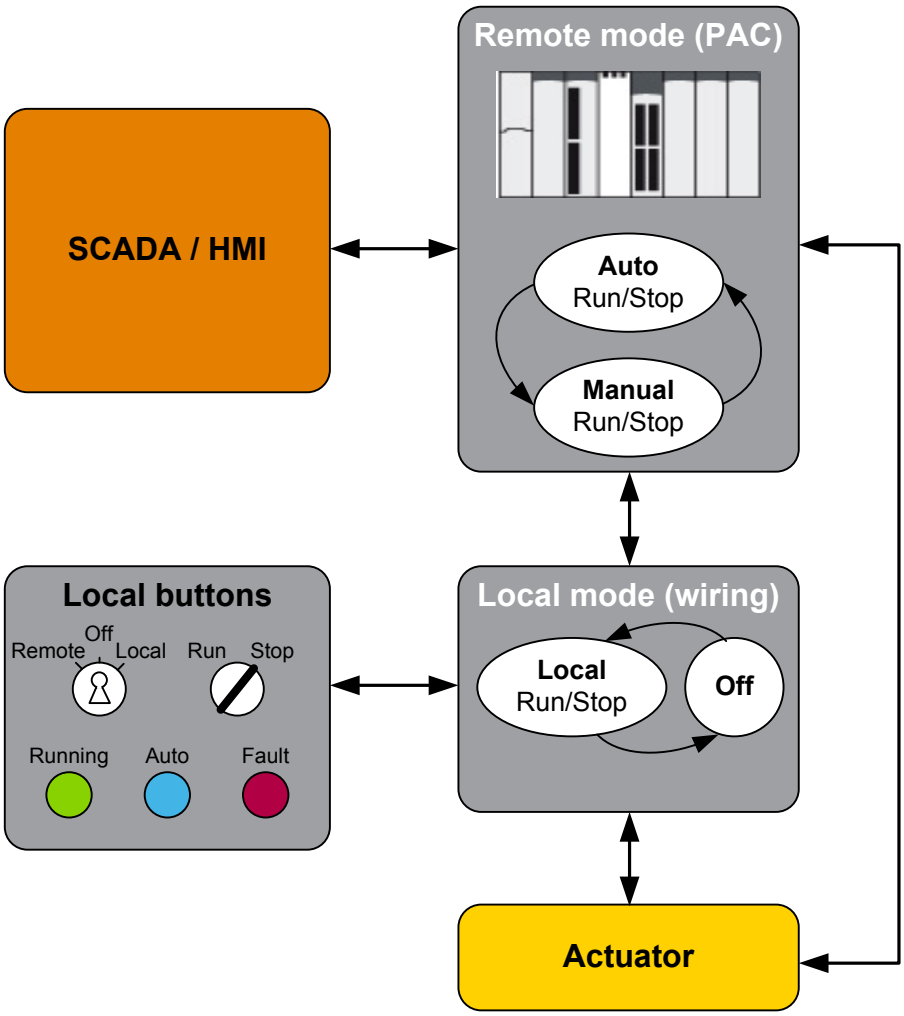
Off Mode

It is possible to switch an actuator to the Off mode for maintenance purposes. This mode prevents any operations on the actuator.

For security reasons, this mode requires an additional electrical padlocking.

Note: Concerning all the modes, the power can be switched off by emergency switches.

The following diagram presents the relation between the different parts of the system.



3.3. Hardware Design

3.3.1. Introduction

The operating modes described above require a wiring design for an emergency stop circuit and motor control device. The following sections provide recommendations for developing a consistent wiring solution.

Safety requirements impose constraints to protect people and the environment. We recommend safety measures for electrical risks that are defined in IEC 60204-1. This standard specifies, in particular, emergency stop operations.

3.3.2. Emergency Stop Circuits Recommendations

Emergency stops are located in cabinets close to the machines and close to the operators. As the use of intermediate relays is prohibited, the solution of a safety function block is essential in the case of multiple stop commands.

To cover most situations, three basic diagrams are proposed.

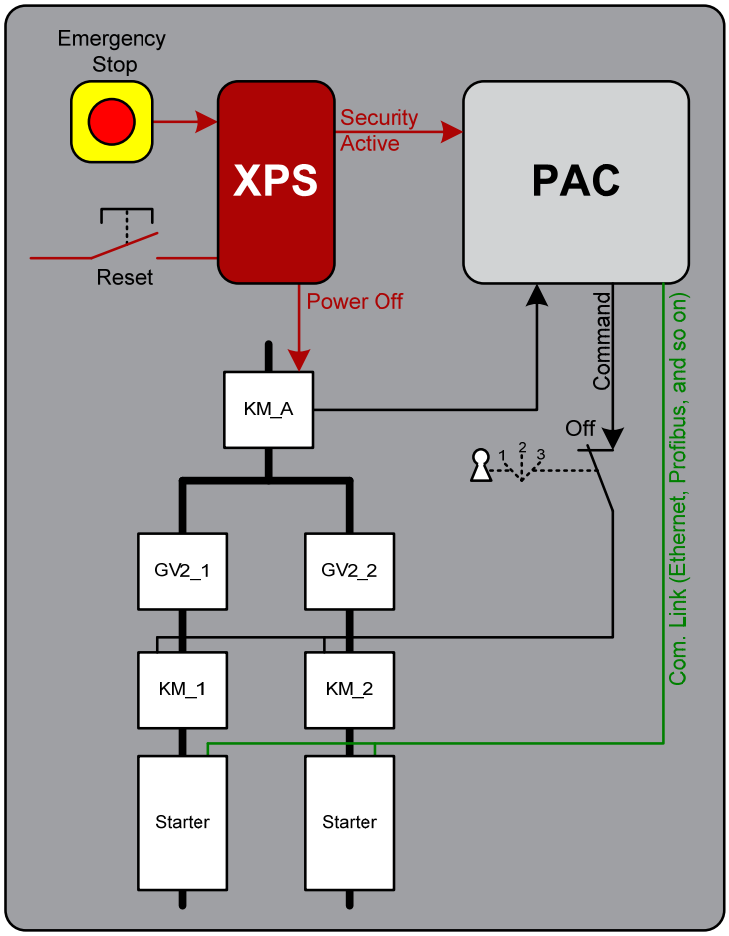
Conventional Diagram

Generally it comprises a contactor, a variable speed drive or a soft starter optionally associated with thermal protection. A second contactor KM_A, in series with KM_1 and 2, makes it possible to cover all categories (conforming to EN ISO 13849-1). The PAC receives information from the safety XPS block and acts on the starter (contactors, variable speed drive....) via the application.

Whatever the mode (local, remote or off), the XPS retains priority. Resetting cannot be performed if KM_1 and KM_2 are closed.

This diagram covers the structures, including the products:

- ATS 48/ATS 22
- ATV 312/ATV 32
- TeSys T
- TeSys D



Note: In the case of the ATV 32, we use a TeSys U to replace the contactor and circuit breaker. Refer to the diagram on the next page for wiring details.

Diagram with TeSys U

A TeSys U integrates a power switching function (represented by KM_1 and KM_2 in the previous conventional diagram) as well as protection functions (short circuit protection, thermal protection and so on).

The standard requires double power breakers; contactor KM_A is mandatory.

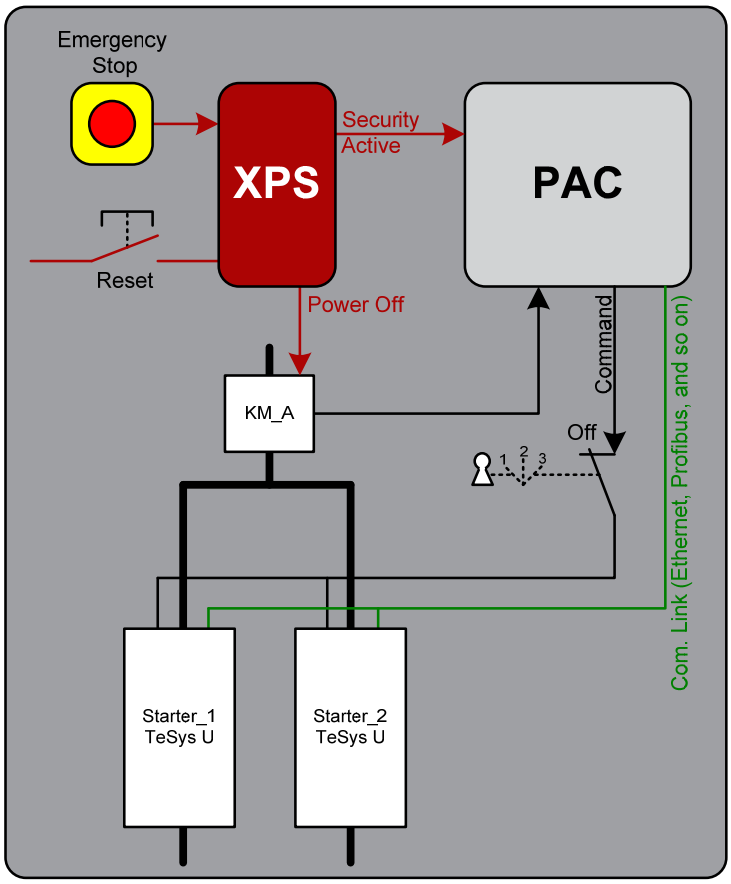
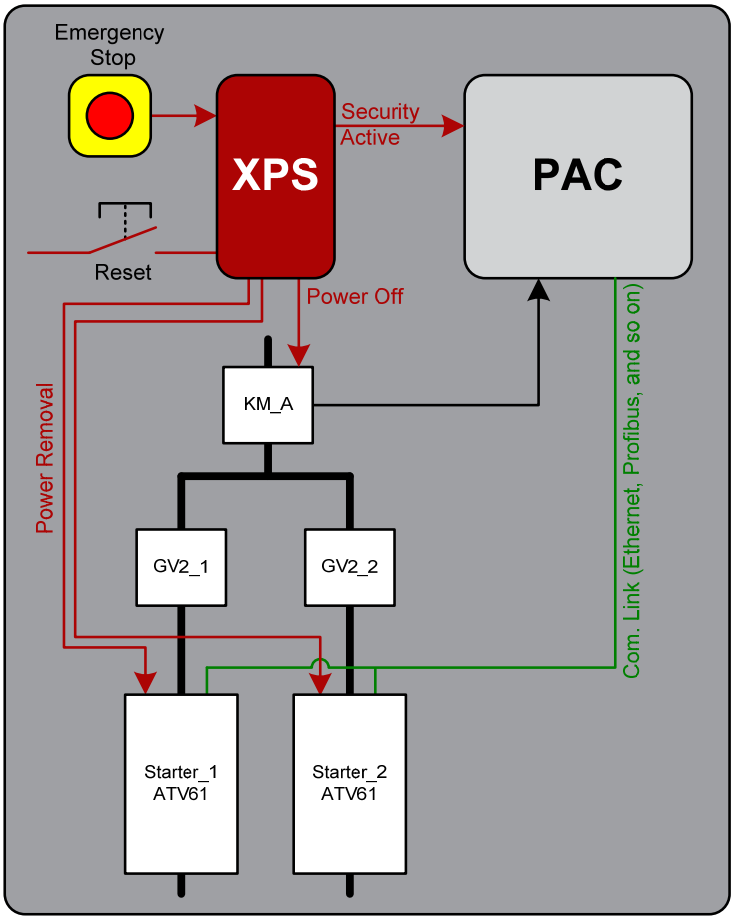


Diagram with Variable Speed Drive ATV61

In the case of a requirement not exceeding level 3 (in ISO 13849-1), a contactor between the GV and the starter is not necessary. The power is shut down by the function Power Removal (PWR) directly wired on ATV 61.



3.3.3. Motor Control Device Wiring Diagram

This section provides a wiring diagram for the following starters:

- Direct On Line starting with starters TeSys U
- progressive starting with soft starters ATS 48
- starting at variable speed with variable speed drive (VSD) ATV61

The proposed diagram re-uses previously defined operation modes (see chapter 3.2 Page 33).

Direct Online TeSys U Diagram

The TeSys U is a starter that integrates sectioning, protection, overload, short circuit and commutator functions. The Remote/Off/Local selector switch allows the TeSys U to be controlled either by the PAC in the Remote mode or by a Run/Stop switch in the Local mode.

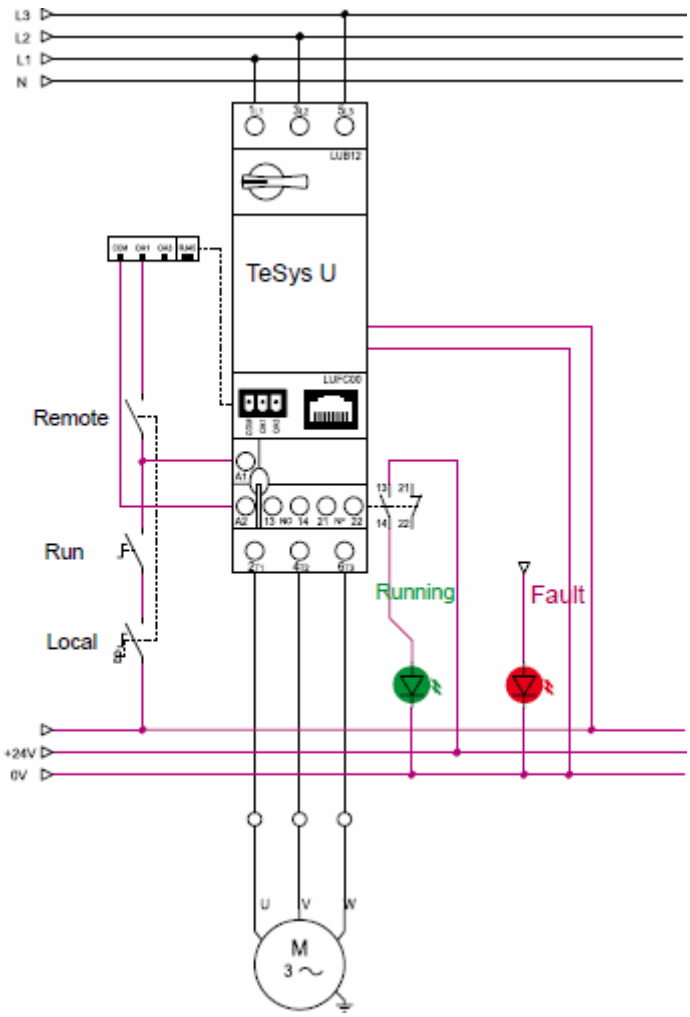
The selector switch has a third position, Off. In this position, the contactor is open and the coil is not powered.

Note: The Off mode cannot be considered as a padlock function.

In the Local mode, the starter is controlled via the Run/Stop selector switch; the Run command is wired directly to the starter. In this mode, the PAC is no longer in the circuit, which is important in the event of spurious signals or for maintenance. Remote and Local positions of the lockable switch selector are transmitted to the PAC.

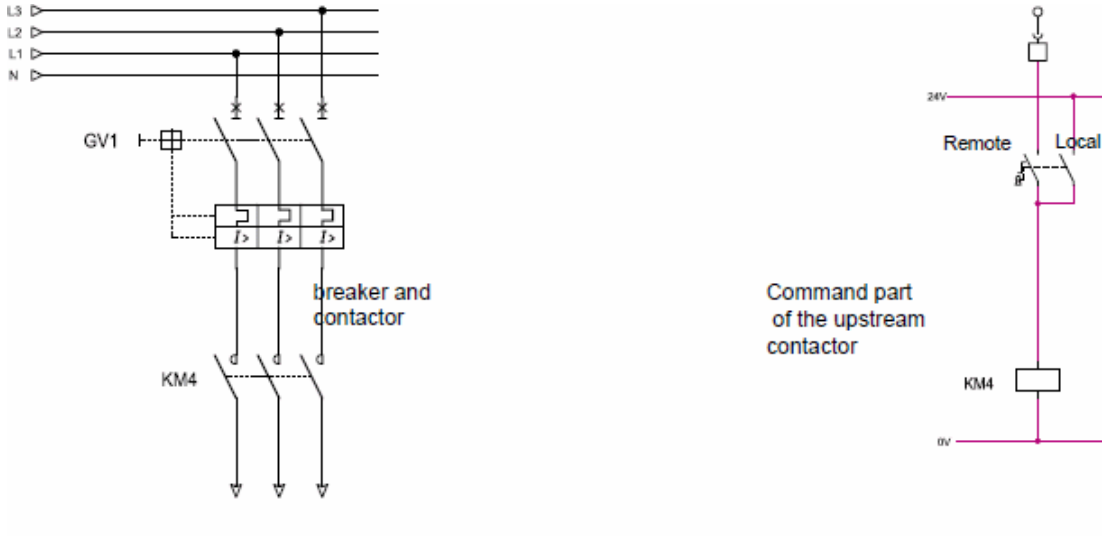
Indicator lights wired to the starter display Run and Fault status on the local panel.

Note: The TeSys U pre-wiring accessories cannot be used in the case of a local command.



Soft Starter ATS 48 Diagram

As described in the following diagram, a KM4 contactor is placed upstream of the ATS 48. This contactor, controlled by the Remote/Off/Local selector switch, allows cutting off the power to the starter (see diagram below).



Note: The Off mode cannot be considered as a padlocking function.

In the Local Mode, the starter is controlled via the Run/Stop selector switch; the Run command is wired directly to the starter and the PAC is no longer in the command circuit. The Remote and Local positions of the lockable switch selector are transmitted to the PAC.

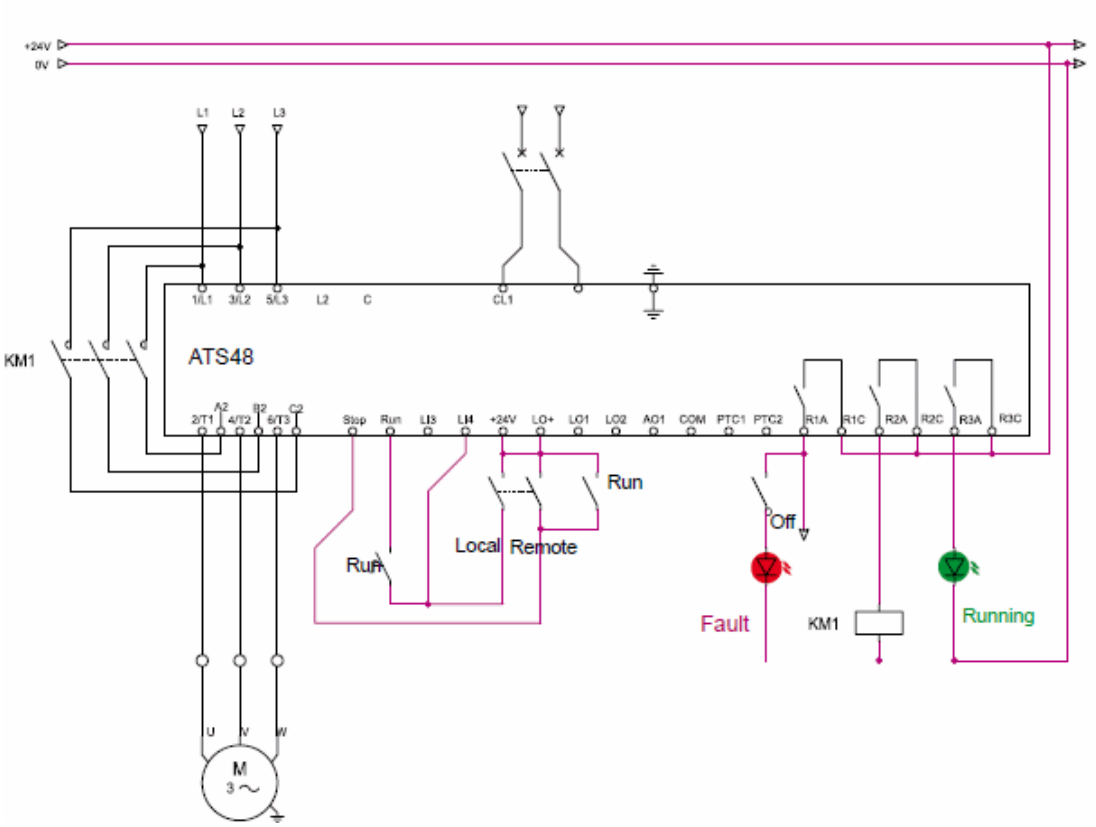
Indicator lights, wired on the PAC or device outputs, display Run, Fault and Auto status on local panel.

A bypass is implemented in order to reduce power consumption. When the start sequence is terminated, the starter is short circuited by a contactor.

In our case, the bypass function is performed by the KM1 contactor, this contactor is controlled via the R2 output of the ATS 48.

Note: To configure the I/O of the starter, PowerSuite (for PC) or ATS Display software are recommended.

The wiring of the ATS 48 is presented on the diagram in the next page.


+24V
0V
L1
L2
L3
1/L1
3/L2
5/L3
L2
C
CL1
KM1
ATS48
2/T1
4/T2
6/T3
Stop
Run
LI3
LI4
+24V
LO+
LO1
LO2
AO1
COM
PTC1
PTC2
R1A
R1C
R2A
R2C
R3A
R3C
Run
Local
Remote
Run
Off
Fault
KM1
Running
U
V
W
M
3

Variable Speed Drive ATV 61 Diagram

As described in the diagram on following page, a KM1 contactor is placed upstream of the ATV 61. This contactor, controlled by the Remote/Off/Local selector switch, allows cutting off the power to the starter.

Note: The Off mode cannot be considered to be a padlocking function.

In the Local Mode, the starter is controlled via the Run/Stop selector switch; the Run command is wired directly to the starter and the PAC is no longer in the command circuit. The Remote and Local positions of the lockable switch selector are transmitted to the PAC.

Indicator lights, wired on the PAC or device outputs, display Run, Fault and Auto status on local panel.

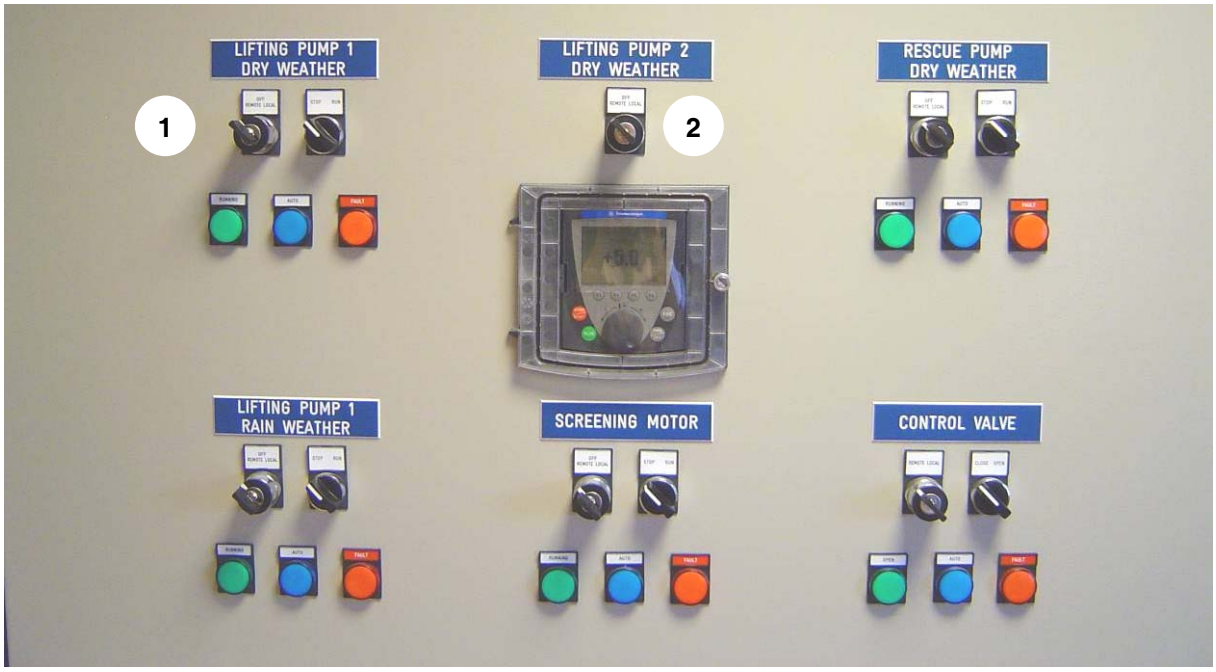
Note: To configure the I/O of the starter, PowerSuite (for PC) or Graphic pocket (Pocket PC) software are recommended.

Depending on the level of security required, a wiring alternative is possible by using the "Power Removal" input on the variable speed drive (see Paragraph 3.3.2 p.36).

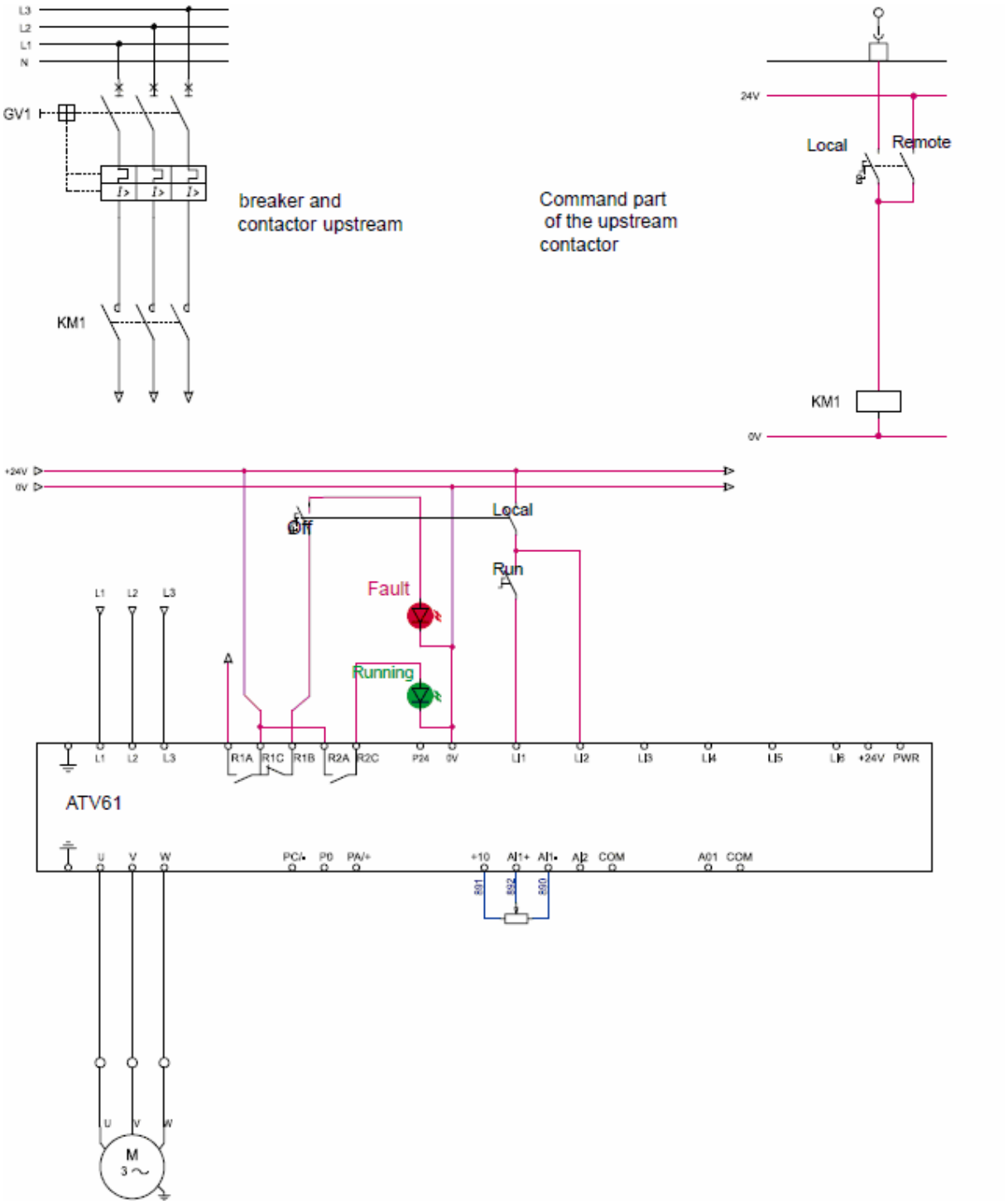
The picture below illustrates the several types of local commands found on the front panel.

1) A **Remote/Off/Local** selector associated with an independent **Run/Stop** selector.

2) A **Remote/Off/Local** selector associated with **Run/Stop** buttons integrated into the remote graphic display terminal.



The wiring of the ATV 61 is presented on the diagram below.



3.4. Software Design

3.4.1. Introduction

Motor control applications require the design of objects dedicated to different devices using the following software:

- PAC application with Unity Pro
- SCADA application with Vijeo Citect

The main components of the architecture need to exchange data and data types during build in order to achieve a consistent application, and to exchange data during run time in order to execute effective and complete process control.

3.4.2. Principle

On PAC Application (Unity Pro)

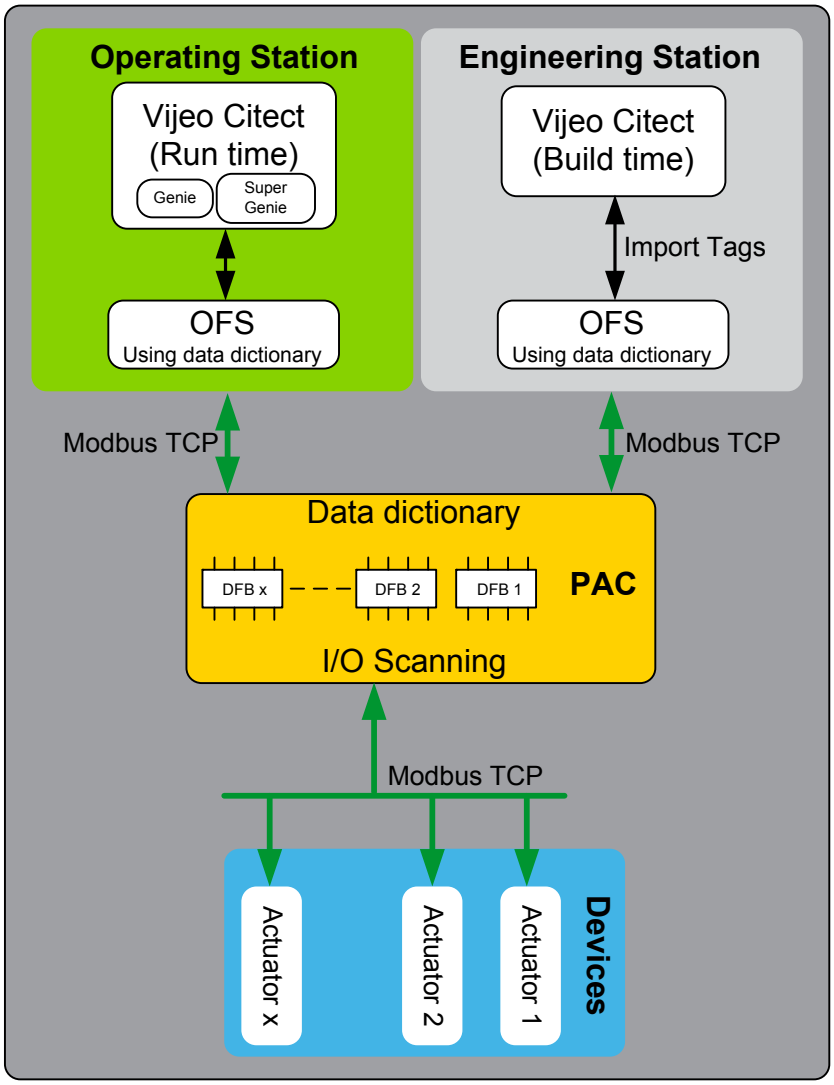
The management of each actuator is handled by Derived Function Blocks (DFBs).

The cyclic exchanges between the PAC and the field devices are managed by IO Scanning.

On SCADA Application (Vijeo Citect)

The HMI related to actuators is based on object-oriented technology (Genies and Super Genies).

The diagram below illustrates the data exchanges between the different components of the system:



3.4.3. PAC Application: DFB Design

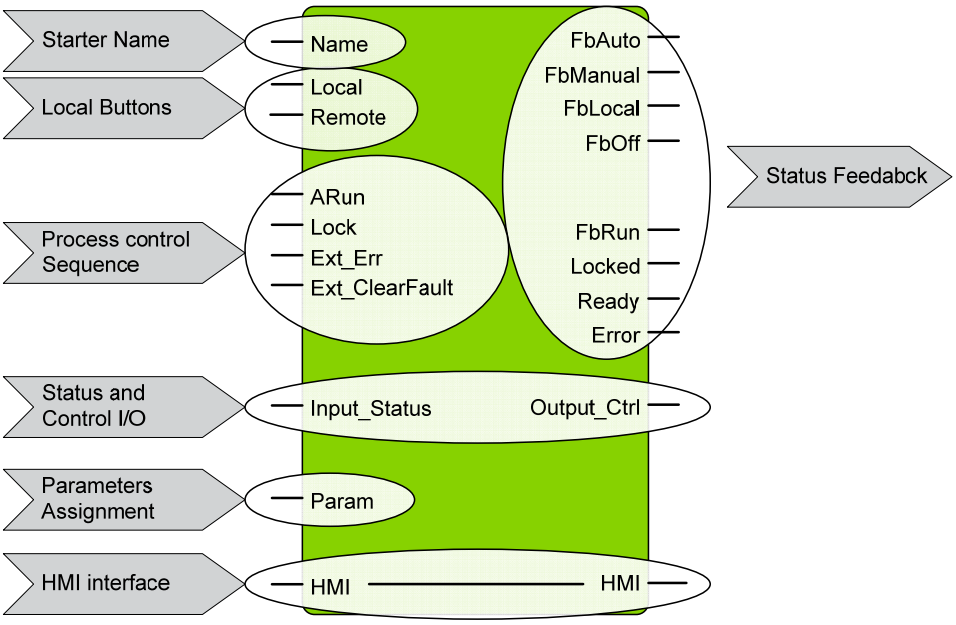
Introduction

The exchanges between the PAC application and motor control devices are managed using DFBs. A single DFB is used for each type of starter and type of network interface. It handles the management of:

- starter name
- operating modes: Local buttons
- link between the sequences and the process status: Process control sequence
- interface with the starter: I/O Status and Control
- adjustment of parameters: Param
- human/machine interface: HMI interface
- status feedback

DFB Structure

The general structure of the proposed DFB interface is described below. In order to have consistent interfaces, the same general structure is applied to all device DFBs.



Note: The detail of the parameters described below represents the general structure of the DFBs, common for all starters. Depending on the starter, the groups may contain additional parameters (speed setpoint, forward/backward control, and so on).

Starter Name

This input pin specifies the starter name used by the SCADA/HMI.

Name	Name of the starter directly used by SCADA/HMI

Local Buttons

Operating mode selection between Remote/Off/Local selector switch position inputs.

Local	Sets the DFB to Local mode. The actuator is directly piloted by the local buttons, the commands from SCADA/HMI are ignored.
Remote	Sets the DFB to Remote mode. The PAC manages the actuator. The commands come from a process sequence in Auto mode or from SCADA/HMI in Manual mode.

Process Control Sequence

This group is the DFB inputs used by the PAC program sequence to control the device in Auto mode.

ARun	In Auto mode, starts the actuator (signal set to 1).
Lock	Interlock input for actuator operation. Actuator operation is stopped or inhibited when the input is set to 1. When signals returns to 0, the actuator is automatically restarted or ready to start depending on the ARun command in Auto mode or HMI command in Manual mode.
Ext_Err	Input for external error signals (emergency stop for example). Actuator operation is stopped or inhibited when the input is set to 1. The error must be acknowledged.
Ext_ClearFault	Input for external error acknowledgement.

Note: The pins Lock (Interlock) and Ext_Err (External Error) inhibit or stop device operation. The main difference is that the Ext_Err needs an acknowledgement via SCADA/HMI before authorizing a new start of the device.

Control I/Os

This group is the process data inputs and outputs (I/Os) that monitor and control the devices. IO Scanning, based on periodic read/write variables, allows implicit data exchanges. This functionality must be reserved for frequently used variables; the access to other variables is performed by explicit data exchanges.

I/O SCANNING INPUT	
Input_Status	Starter status input. Feedback signal from I/O Scanning.

I/O SCANNING OUTPUT	
Output_Ctrl	Starter control output. Sent to the starter via I/O Scanning.

Parameters Assignment

The input data type is a Derived Data Type (DDT) containing at least the following parameters:

Discrepancy_time	Max time between the order and the order's feedback. If max time is reached, a discrepancy error is generated.
Nb_Start_Day	Number of starts authorized per 24 hours.
Min_time_Stop	Minimum time between stop and restart.

These DFB input parameters are adjusted during the start-up of the installation by assigning constant values to the variables. In our case, some of these parameters are sent to HMI or SCADA as read only.

As these parameters are not defined as constants, they are not saved in the PAC memory. All modifications applied to these parameters during operation are lost on a cold-start.

To overcome this loss, the following procedure can be applied to save the current parameter values as initial values on a PAC cold start:

Step	Action
1	Using Unity Pro, validate the **save** attribute of all device parameter variables.
2	Adjust device parameter values using Unity Pro.
3	Using Unity Pro, in the connected mode, set the system bit %S94 to 1: Initial values of all variables with the **save** attribute are replaced by their current values.
4	Save the application. **Note:** On a M340, it is necessary to set the system bit %S66 to 1 or to transfer the RAM application to the memory card.

HMI Interface

The input data type is a Derived Data Type (DDT) containing at least the following variables. These variables are exchanged between the PAC and SCADA/HMI.

Aut_Man	Set the block in Automatic (true) or Manual (false) mode.
Run_Stop	Run (true) or stop (false) the actuator.
Clear_Fault	Acknowledge errors indicated at the output Error.
Nb_Start	Number of starts performed in the last 24 hours.
Time_BStartStop	Time (seconds) between start and stop of the actuator .
Discrepancy_time	Max time between the order and the order's feedback. If max time is reached, a discrepancy error is generated.
Time_to_Start_s	Time (seconds) before the next start.
Time_to_Start_m	Time (minutes) before the next start.
Sts	Status structure
├Auto	Automatic mode is activated. The process sequence manages the actuator.
├Manual	Manual mode is activated. SCADA/HMI manages the actuator.
├Local	Local mode is activated. Hardwired selector switches manage the actuator.
├Off	Off mode is activated. The starter power is cut off.
├Ready	Starter ready to start.
├Run	Starter is running.
├Warning	Starter in warning.
├Error	Starter in error.
├Locked	Starter is locked by an interlock (Lock input).
├TimerProtect	Starts number monitoring is activated and the max starts number is reached. Monitoring enabled if Nb_start_day≠0.
└Discrepancy_Err	Discrepancy monitoring is activated and the discrepancy time is reached. Monitoring enabled if Discrepancy_time≠0.
Name	Name of the starter

Status Feedback

These output variables indicate the status of the device. They can be used in the PAC process application and also to connect a PAC digital output.

FbAuto	Auto mode activated
FbManual	Manual mode activated
FbLocal	Local mode activated. The actuator is managed by local switch selector
FbOff	Powered off by local switch selector
FbRun	Starter is running
Locked	Starter is locked
Ready	Starter is ready
Error	Starter in Error

Operation

DFBs support Automatic and Manual operating modes. The Automatic and Manual modes are activated by the SCADA/HMI when the mode is set to Remote. The Manual mode is selected by default on a PAC cold start. The Local mode information from hardware selector is connected on an input pin to inhibit the commands from the process and SCADA/HMI. This information is also displayed on SCADA/HMI.

In the Automatic mode, the motor is started and stopped via the inputs ARun in Remote mode only. If the operating mode is changed from Automatic to Manual, the motor continues in the same way - run to run, stop to stop. If the operating mode is changed from Manual to Automatic, the motor follows the commands from the process.

If the operating mode is changed from Remote to Local, the motor stops but the local hardware command leads the stop or the run on starter terminal block. If the operating mode is changed from Local to Remote, the motor follows the commands from the process or from the HMI.

If the interlock input Lock is set to 0, the motor is running. An active interlock signal inhibits the start of the motor or stops a running motor. The motor restarts or is ready to start when the interlock signal returns to 0.

The DFB sets the output Error signal to 1 if the error input Ext_Err is set to 1 (external error) or in the case of an invalid operating mode, a missing feedback signal or an internal error of the starter. The errors are displayed on the HMI as alarms. To reset the output Error, an acknowledgement must be done by a rising edge on the input Clear_Fault from the HMI structure or on the input Ext_ClearFault from the PAC application.

A minimum stop time before a new start of the motor can be set using the Min_time_stop parameter. The motor is then authorized to restart only when the Time_to_start = 0. This function is disabled if Min_time_stop = 0.

Also, a maximum number of starts per day can be set using the Nb_Start_Day parameter. The motor is then authorized to restart as long as Nb_Start $\leq$ Nb_Start_Day. This function is disabled if Nb_Start_Day = 0.

3.4.4. SCADA Application: System Design

Introduction

During the SCADA application build, Vijeo Citect objects have to be defined as well as the data and data type associated.

The following recommendations are provided to facilitate design, readability and re-use:

- Exchanges are mainly performed via DDT variables (Derived Data Type).
- OFS server (OPC Factory Server) allows use of Unity Pro structured variables in unlocated format.

SCADA System Objects

The application is based on Genie and Super Genie objects.

Genie and Super Genie are useful when there are many devices of the same type. They can be re-used many times without re-configuring them for each device. Once the common information is configured, the device-specific information is passed to the Genie and Super Genie at runtime.

Genies and Super Genies can be created in an another project and then, re-used easily by a new Vijeo Citect project using the function “Included Projects”.

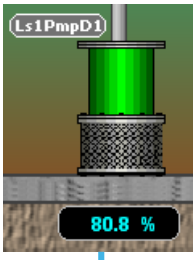
SCADA System Example

Genie and Super Genie objects are associated to each type of starter element.

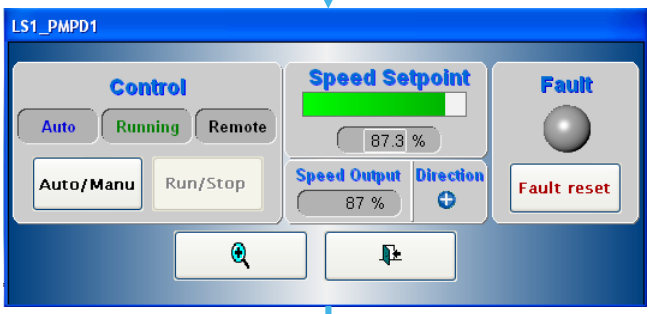
A Genie represents a simple object such as the Ls1PmpD1 pump in the next figure.

A Super Genie is a dynamic page able to exchange tags related to the object it has been called from. Generally, a Super Genie is attached to a Genie.

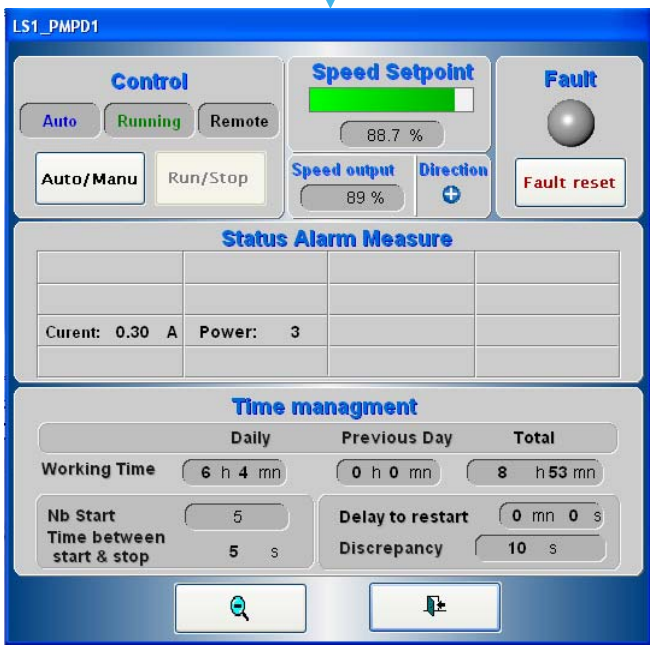
Genie



Super Genie 1



Super Genie 2



4. Configuration

4.1. Introduction

The aim of this chapter is to provide information for configuring the various components of the system (PAC, motor control devices and SCADA applications). The main purpose is to build a consistent system configuration with the description of all data exchanged between key solution components.

The configuration of the system comprises the following stages:

- In our case, the data exchanges between the PAC and the field devices require the configuration of the 2 following services:
  - configuration of the IO Scanning communication service
  - configuration of the Faulty Device Replacement (FDR) service for the relevant devices

  The configuration of the FDR service is performed before the IO Scanning configuration because we want to first set up the device names and the associated IP addresses.
- Configuration of data and data types used both in PAC and SCADA applications. A recommendation is provided to allow a unique configuration of data in both tools.

4.2. PAC and Motor Control Device Configuration

This chapter describes the configuration of the FDR and IO Scanning services.

4.2.1. FDR Service

Principle

This service is based on identification of the device by a Device name. Each Device name must be unique over the Ethernet network.

The FDR server controls duplication of Device names and does not assign an IP address that has already been assigned and is active.

The new device (FDR client) retrieves:

- its IP address and the FDR file path from a DHCP server
- the FDR file from an FTP server, if the device is not configured in local configuration

Note: The DHCP server and the FTP server are the same device (TSX Premium or Quantum PAC).

The FDR file contains the Ethernet parameters (configuration of IO Scanning, FDR, and so on) and the device parameters.

3 configuration steps are required prior to becoming operational:

- device name definition
- clients (field devices) configuration
- server (PAC) configuration

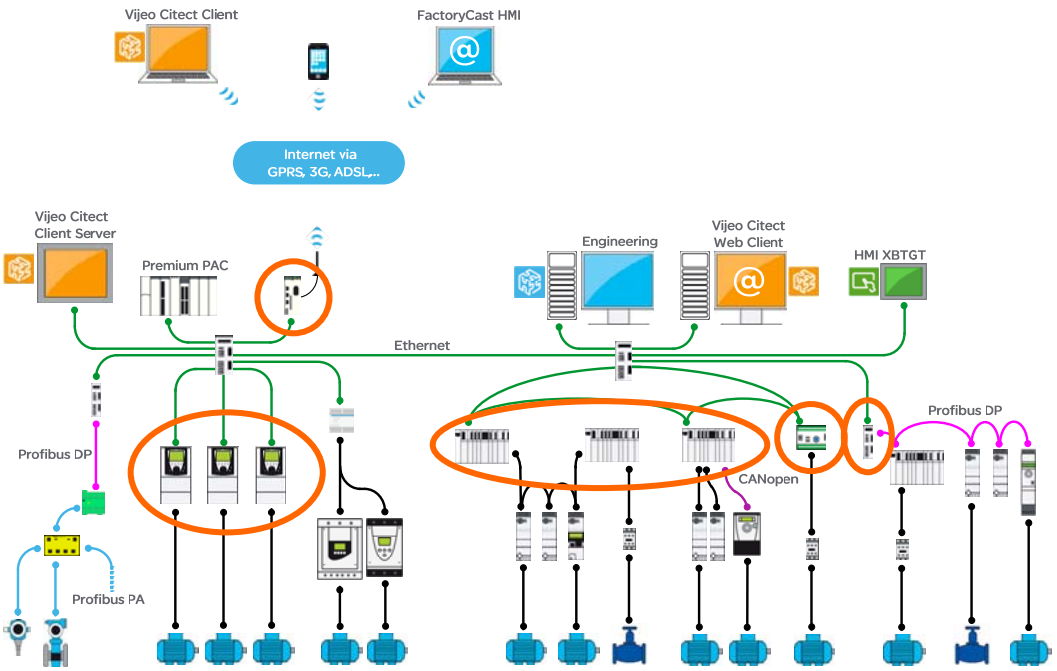
Device name definition

The device name allocations depend on the device configuration methods:

- rotary switches
- graphic display terminal
- Web server
- PowerSuite tool

In our architecture, 5 types of devices handle the FDR service:

- ATV61 variable speed drive
- Distributed I/Os modules Advantys STB NIP 2311
- TeSys T controller
- Profibus Remote Master (PRM) gateway module
- FactoryCast Gateway ETG3021



For our application, the **Device names** are allotted as described in the table below:

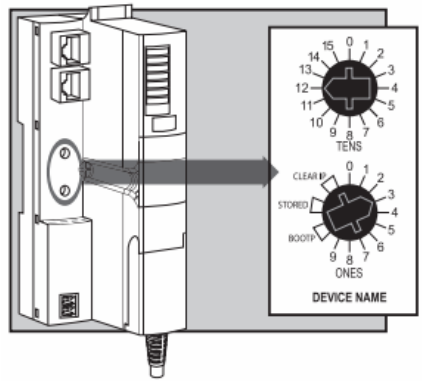
Device	Device Name
Advantys STB NIP 2311 Lifting & Screening Unit	STBNIP2311_020
Advantys STB NIP 2311 Grease and Sand Removal Unit	STBNIP2311_021
TeSys T Air Compressor Grease and Sand Unit	TeSysT001
ATV61 Pump 1 - lifting wastewater tank	ATV61_PmpD1
ATV61 Pump 2 - lifting wastewater tank	ATV61_PmpD2
ATV61 Pump 3 - lifting wastewater tank	ATV61_PmpD3
PRM - Primary Clarifier Unit	TCSEGPA23F14F001
ETG3021 - wastewater station	ETG3021_WATER

Clients Configuration

By default, each device has a Device name that can be customized if necessary. In our case, the default names are used for the devices equipped with rotary switches.

Advantys STB NIP 2311

The default Device name is formatted as follows: **STBNIP2311_xxx**, where xxx is defined by the rotary switches located on the STB NIP 2311 front panel.

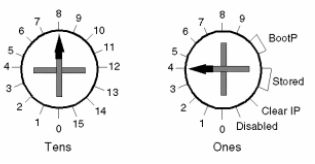


For the first STB Lifting – Screening, the Device name has to be set to **STBNIP2311_020.** Set the Tens switch on “2” and the Ones switch on “0”.

For the second STB Grease and Sand Removal, the Device name has to be set to **STBNIP2311_021.** Set the Tens switch on “2” and the Ones switch on “1”.

TeSys T controller

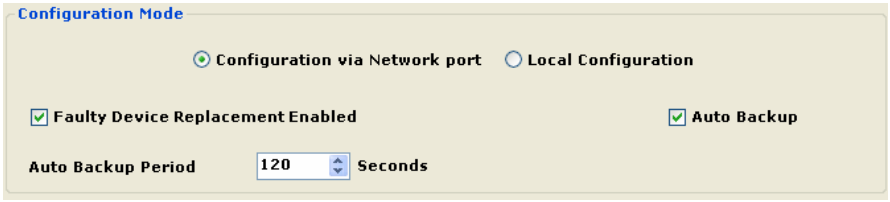
The format of the Device name is TeSysTxxx where xxx is defined by the value of the rotary switches located on the front panel of the TeSys T.



- To set the Device name to **TeSysT001**, set the Tens switch on “0” and the Ones switch on “1”.

Then, the PowerSuite software is used to configure the FDR service.

- From **Settings→Communication and HMI** in **Configuration Mode** area, select **Configuration via Network port**, **Faulty Device Replacement Enabled** and **Auto Backup** (automatic periodic backup of the FDR configuration file to the FDR server) as described in the figure below.



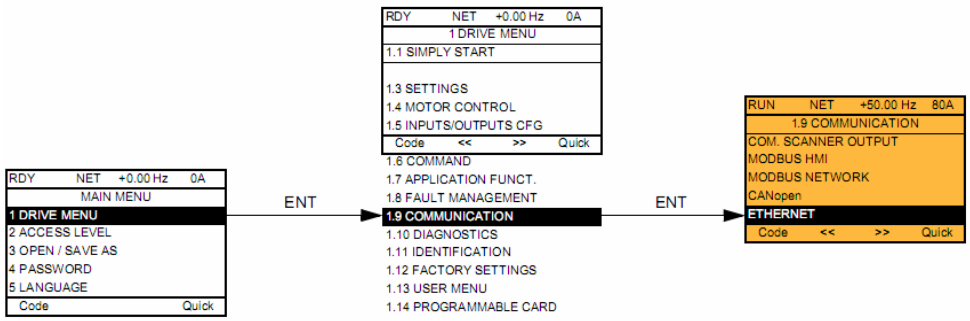
Note: By default, the **Configuration via Network port** and **Faulty Device Replacement Enabled** parameters are selected, allowing a replacement of the device without using PowerSuite software.

ATV 61 variable speed drive

For the ATV 61, the Device name is set using the “graphic display terminal” (Power Suite software can also be used, but in that case, a connection to the device is required).

The first wastewater lifting pump is used here as an example:

From the “graphic display terminal”, access to the Ethernet menus as described in the diagram below.



From the Ethernet menu, the following actions are performed:

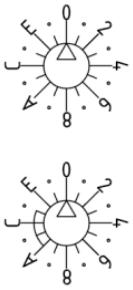
Step	Action
1	Select the menu **DEVICE NAME** and type “ATV61_PmpD1” to set the Device name.
2	Select the menu **FDR validation** and select **Yes** to activate the FDR service.
3	From the menu **FDR autosave**, select **Yes** to save the configuration automatically.

Note: The **IP Card address** must be set to “0.0.0.0” to run the DHCP service.

Proceed in the same manner for the 2 other ATV 61s using Device names **ATV61_PmpD2** and **ATV61_PmpD3**.

PRM

The format of the Device name is **TCSEGPA23F14Fxxx** where xxx is defined by the rotary switches located on the back panel of the PRM.



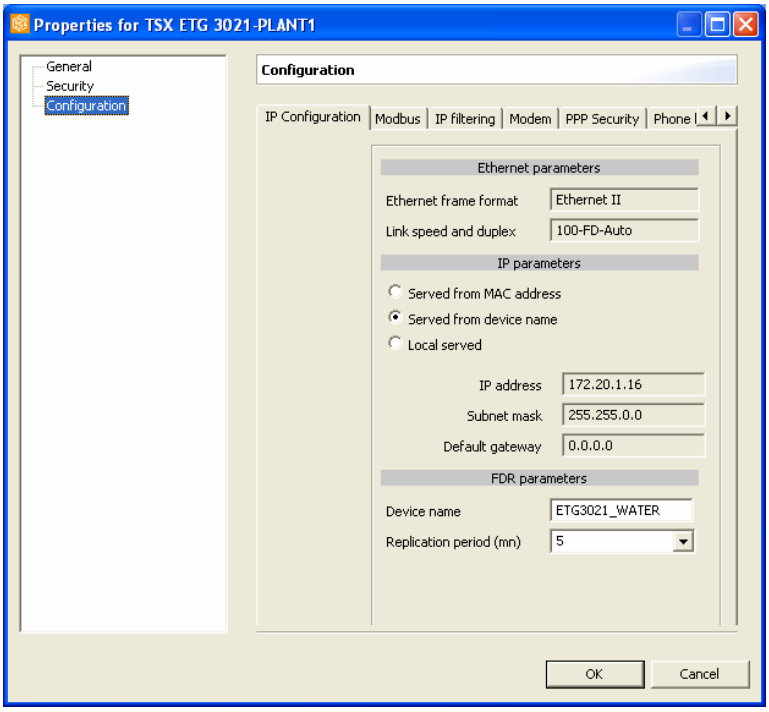
For the PRM, to set the Device name to **TCSEGPA23F14F001**, set the Tens switch (top one) on “0” and the Ones (bottom one) switch on “1”.

FactoryCast Gateway ETG3021

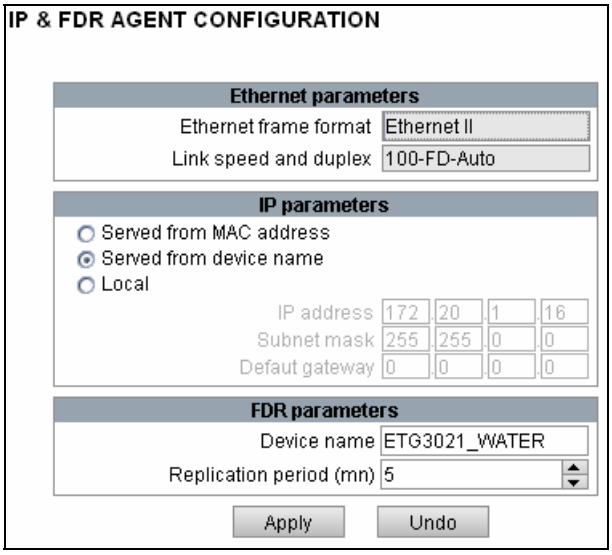
The configuration can be done either from Web Designer or from the ETG3021 web server.

From the ETG3021 **IP Configuration** menu, in the IP parameters area, select **Served from device name.** Then, in the **FDR parameters** area, type "**ETG_3021_WATER**" in the **Device name** field. The replication period parameter sets the automatic backup of the FDR configuration file to 5 minutes.

Configuration from Web Designer



Configuration from the web server



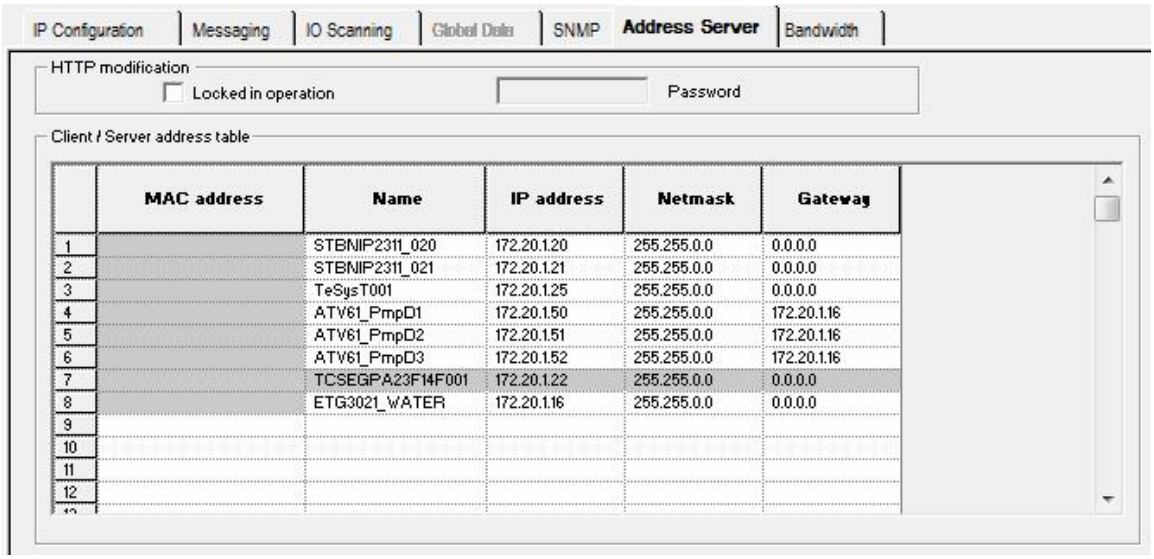
Server Configuration

In the Water application, the DHCP and FDR server are hosted in the Premium PAC TSX P57 4634. The IP address used for each FDR device is presented in the table below.

Device	Device Name	IP
Advantys STB NIP 2311 Lifting & Screening Unit	STBNIP2311_020	172.20.1.20 / 255.255.0.0 / 0.0.0.0
Advantys STB NIP 2311 Grease and Sand Removal Unit	STBNIP2311_021	172.20.1.21 / 255.255.0.0 / 0.0.0.0
TeSys T Air Compressor Grease and Sand Unit	TeSysT001	172.20.1.25 / 255.255.0.0 / 0.0.0.0
ATV61 Pump 1 - lifting wastewater tank	ATV61_PmpD1	172.20.1.50 / 255.255.0.0 / 172.20.1.16
ATV61 Pump 2 - lifting wastewater tank	ATV61_PmpD2	172.20.1.51 / 255.255.0.0 / 172.20.1.16
ATV61 Pump 3 - lifting wastewater tank	ATV61_PmpD3	172.20.1.52 / 255.255.0.0 / 172.20.1.16
PRM - Primary Clarifier Unit	TCSEGPA23F14F001	172.20.1.22 / 255.255.0.0 / 0.0.0.0
ETG3021 - wastewater station	ETG3021_WATER	172.20.1.16 / 255.255.0.0 / 0.0.0.0

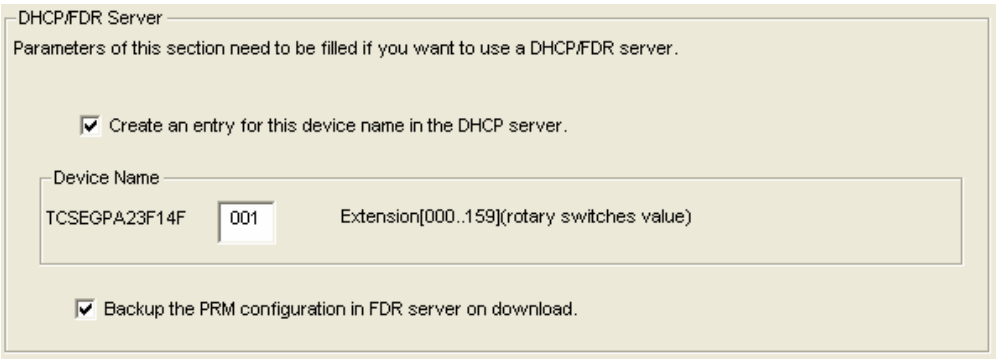
Step	Action
1	Open the project Water from Unity Pro and click on the **Address Server** tab from the **Network Configuration** window.
2	Type the previously defined Device names in the field **Name** and add the corresponding **IP address**, **Netmask** and **Gateway**.

The screenshot below illustrates the procedure:



For the PRM of the Primary Clarifier unit, the configuration of the IP address and Device name are not done from the **Address Server** tab. These parameters are set from the **PRM configuration** screen as described in the procedure on next page.

Step	Action
1	Open the DTM browser from the Tools menu.
2	From the DTM browser, double click on PRM_Master_Clarifier to access the configuration tool.
3	From the menu, select **General Settings**.
4	In the DHCP/FDR Server area, select the boxes **Create an entry…** and **Backup the PRM configuration…**
5	Type **001** in the Device Name field to set the name of the PRM module to **TCSEGPA23F14F001**. This corresponds to the name defined with the rotary switches in the previous section.


DHCP/FDR Server
Parameters of this section need to be filled if you want to use a DHCP/FDR server.
Create an entry for this device name in the DHCP server.
Device Name
TCSEGPA23F14F
001
Extension[000..159](rotary switches value)
Backup the PRM configuration in FDR server on download.

Once the Ethernet configuration is done, build the program and transfer it to the PAC.

Note: It is necessary to restart the devices to finalize their FDR configuration.

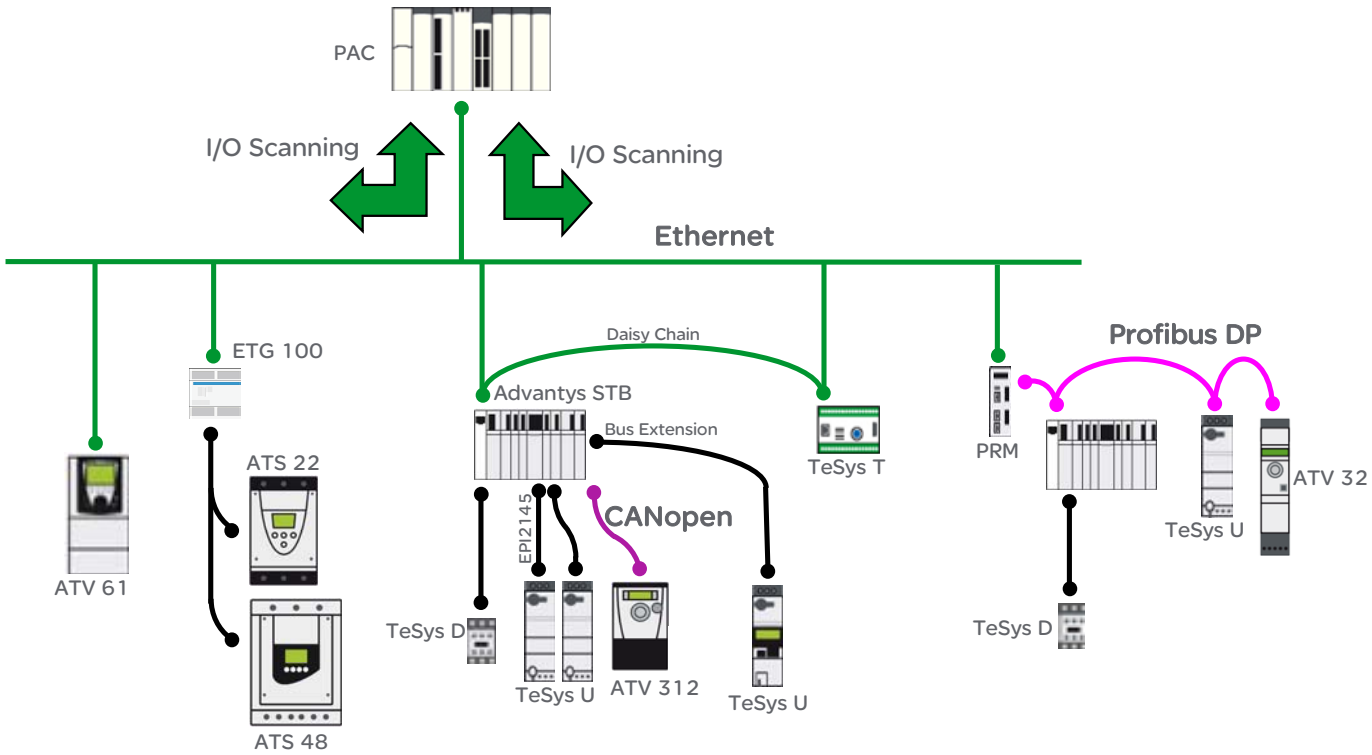
4.2.2. IO Scanning Service

On each scan time, the PAC application exchanges input and output data with the motor control devices.

The IO Scanning communication service is used on the Ethernet network to perform this periodic communication.

To define the IO Scanning service configuration in Unity Pro, it is necessary to identify the data exchanged with each motor control device. These data will be used by dedicated DFB (see DFB list chapter 5.2 P.98). In the selected architecture described in chapter 2.5.3 P.30, 4 types of communication are used. The different characteristics in IO Scanning parameters are:

- Motor control devices connected on Advantys STB on Ethernet (TeSys U, TeSys D and ATV 312)
- Motor control devices connected on Profibus DP with a PRM (TeSys U, TeSys D and ATV 32)
- Motor control devices connected behind an Ethernet gateway to Modbus serial line (Soft starter ATS 48 or ATS 22)
- Motor control devices connected directly on the Ethernet network (ATV 61 and TeSys T)



4.2.3. TeSys U configuration

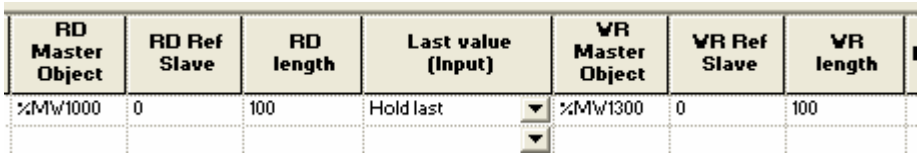
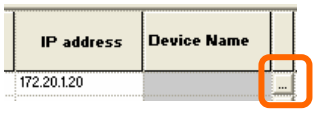
In this case, the TeSys U is connected to the Ethernet network through an Advantys STB island. The connection can be done either using a pre-wiring solution (module Advantys STB EPI2145) or an Advantys extension bus communicator.

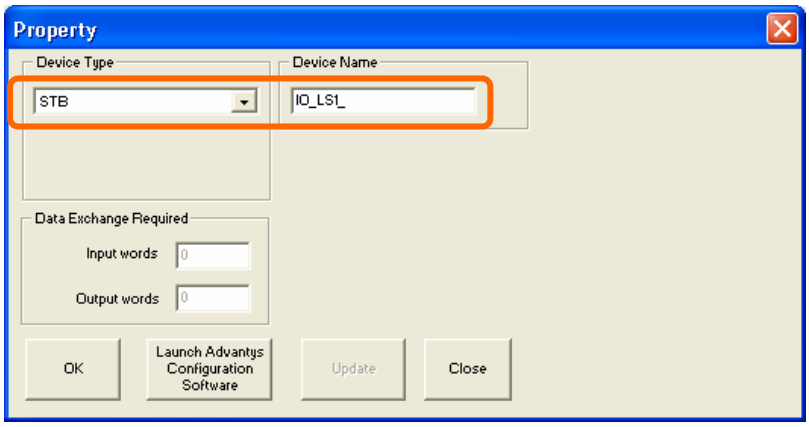
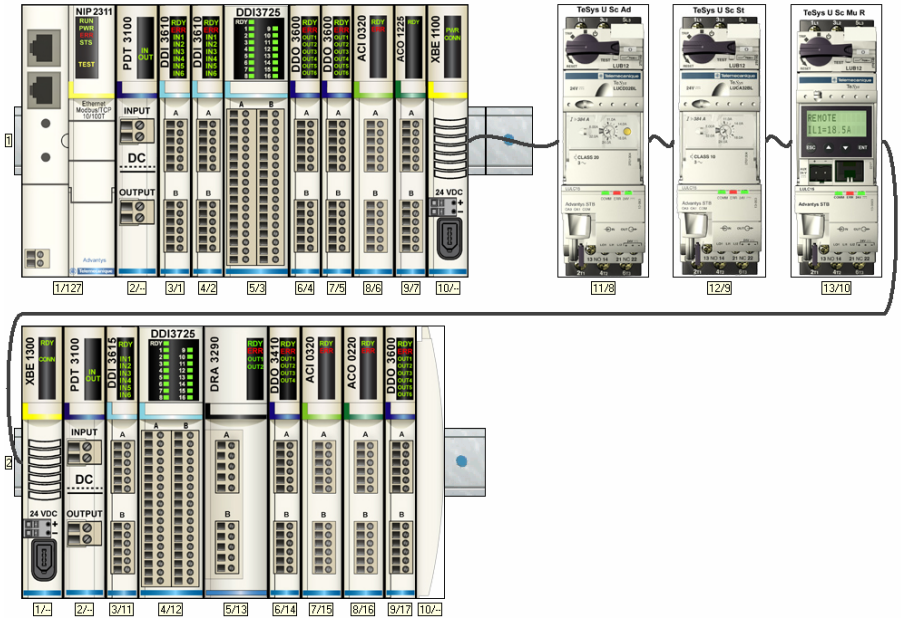
The configuration of motor starter used behind an Advantys STB requires two steps:

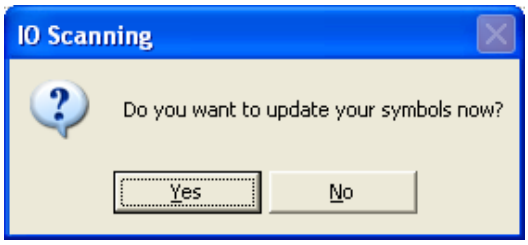
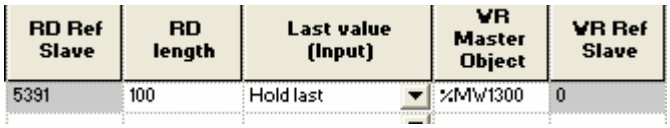
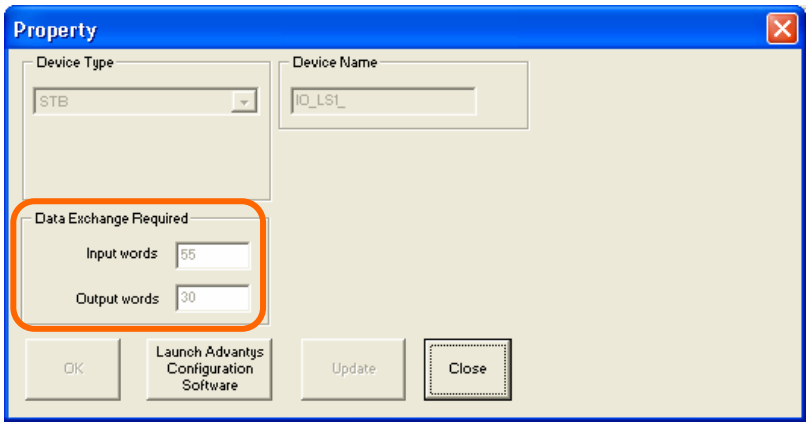
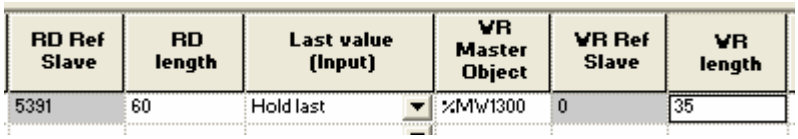
1) The communication between the Advantys STB island and the PAC is configured in IO Scanning.
2) The data of the selected motor starter is identified inside the Advantys STB Island.

IO Scanning configuration

The Lifting Screening STB island (172.20.1.20) is used to illustrate the configuration steps.

Step	Action
1	Run Unity Pro and open the Water project. Click on IO Scanning tab from the Network Configuration window.
2	On a new IO Scanning line, type the island IP address 172.20.1.20 in the IP address field.
3	On the same line, enter the start address (Master Object) and the number of words (length) for the read and write registers: • %MW1000 and 100 words for read register. • %MW1300 and 100 words for write register.  **Note:** It is necessary to configure the RD length and WR length values before running the Advantys tool from Unity Pro but these values are not yet defined. Therefore, the read/write word lengths are overestimated at first and are fine-tuned in step 10.
4	Click on the button in the Device Name field to open the Property window. 

5	Select STB in the combo box Device Type and configure the Device Name to IO_LS1_ **Note**: This Device Name is not the same as the Device Name set in the Address Server tab in the network configuration.  **Note:** The Device Name is used as a prefix to compose the variable names related to this island.
6	Click on the button **Launch Advantys Configuration Software** to run the Advantys software.
7	The Advantys software opens a blank configuration project named IO_LS1_. The island has to be configured at this stage with the I/O modules either by selecting by hand the modules in the catalog, or by uploading the configuration. Refer to the Advantys software documentation for the different island configuration methods.
8	The Lifting and Screening Advantys island configuration is presented in the figure below. 
9	Once the configuration is done, close Advantys software and return to Unity Pro.

10	A dialog opens and proposes to update the symbols linked to this island.  Click on **Yes** to start the update. Read and write reference registers of the island are automatically filled: 5391 for the read register and 0 for the write register. These values cannot be modified and are grayed. 
11	To refine the read and write register number, click again on the button in the Device Name field to open the Property window. Write down the values in the Data Exchange Required area, for this island, 55 input words and 30 output words.  **Close** the window.
12	Set the RD and WR lengths values to 60 and 35 respectively.  **Note:** The values entered are slightly more than necessary to be able to add a module without reloading the network configuration in the PAC.

The Lifting and Screening Advantys STB is now configured and ready to communicate with the PAC.

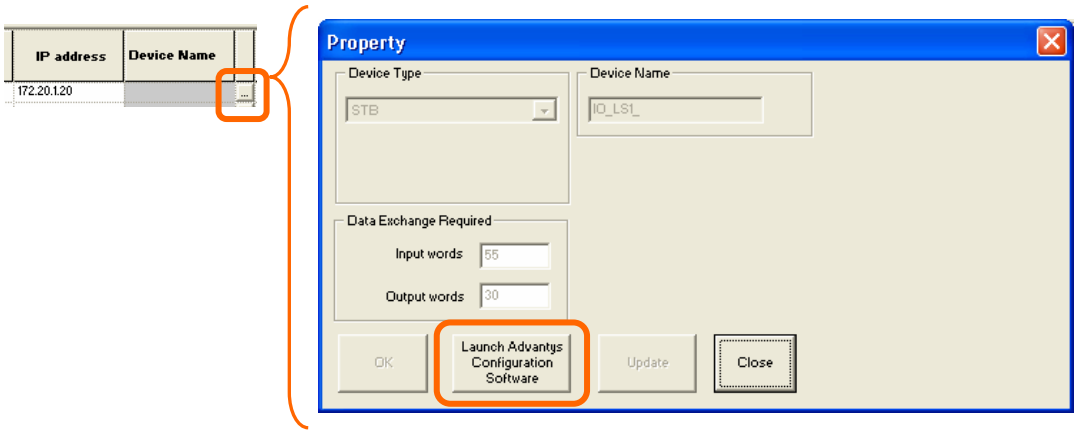
The table below summarizes the mapping created in the IO Scanning for this Advantys STB island.

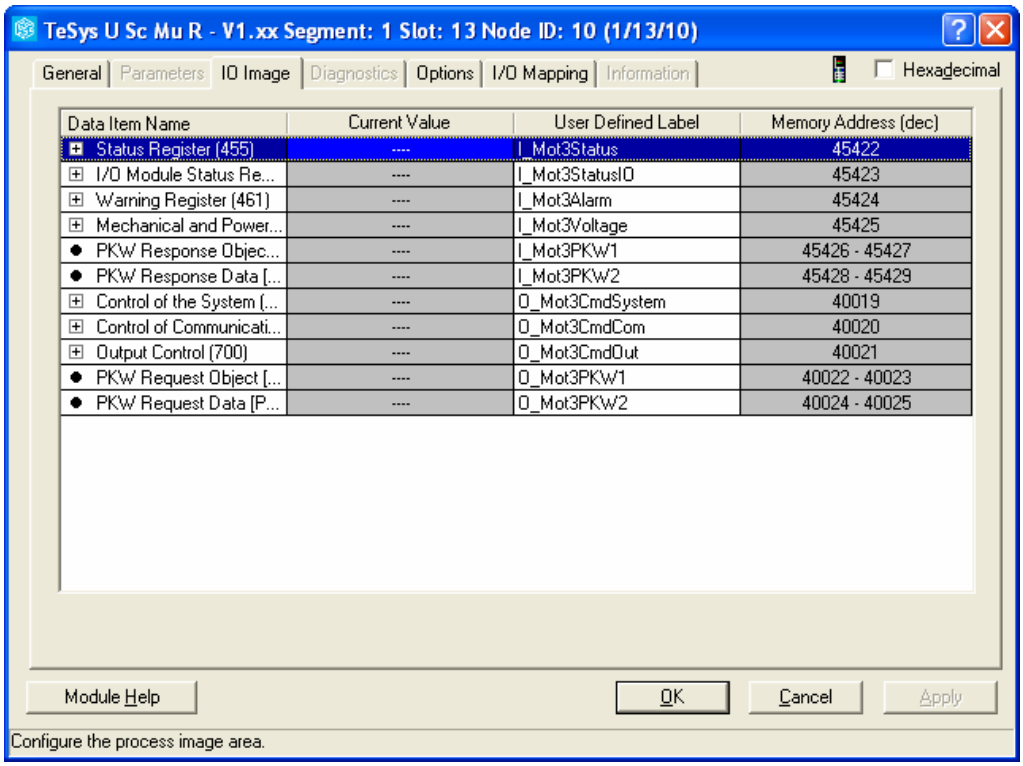
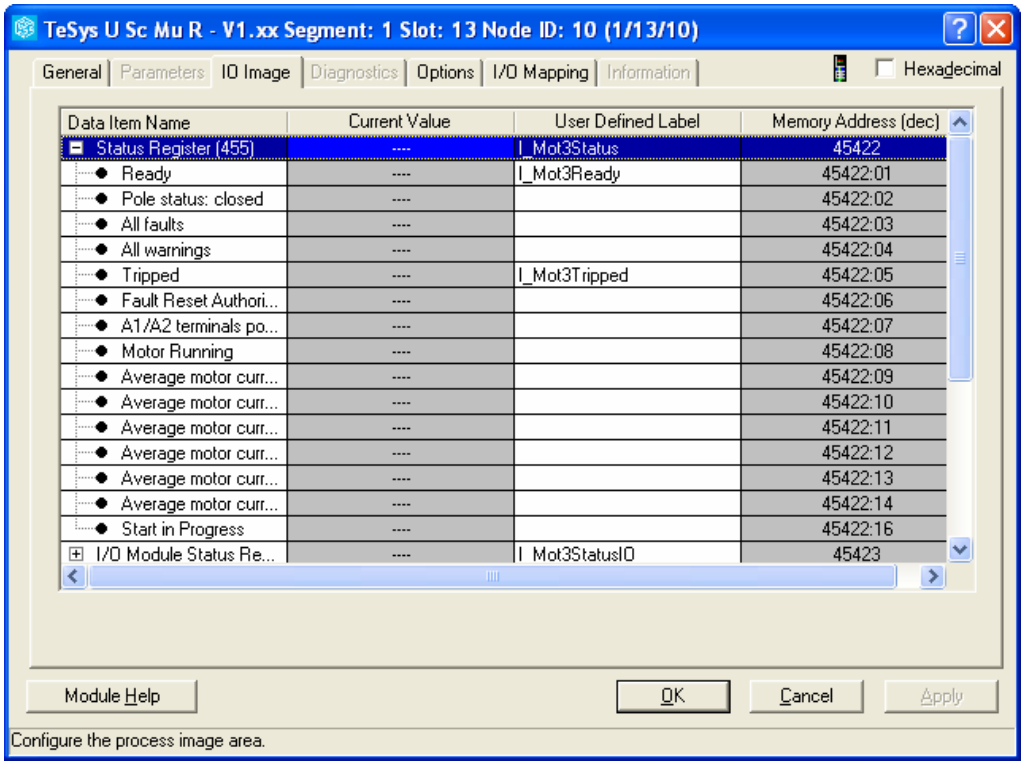
	ADVANTYS		UNITY PRO		
	Modbus Register	Item	RD Slave index	RD Master object	Length
Input data	45392	First island register input	5391	%MW1000	55
	45446	Last island register input	5445	%MW1054	
Output data	40001	First island register output	0	%MW1300	30
	40030	Last island register output	29	%MW1329	

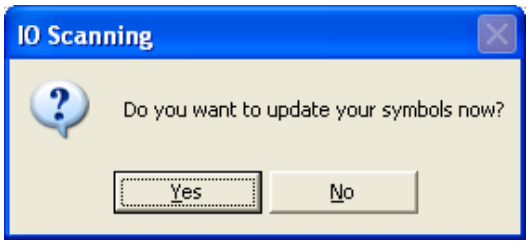
Note: The configuration files of this Advantys STB island generated by the Advantys software are located in a directory defined in the Unity Pro options. Therefore, to use the Unity Pro application on another computer and make some modifications concerning the STB island mapping, it is necessary to copy the Unity Pro application and also the Advantys configuration files in the proper directory of the new computer.

Define the data of the TeSys U

We use the TeSys U Multifunction of the screening unit compactor named Mot3 as an example to illustrate the data definition.

Step	Action
1	Open the Lifting and Screening unit Advantys STB island configuration to define the data exchanged with the chosen TeSys U.
2	From the IO Scanning service, click on the button in the Device Name **IO_LS1_** field to open the Property window of this device.
3	Click on the button **Launch Advantys Configuration Software.** 
4	To be able to modify the TeSys U configuration, unlock the island configuration by clicking on the following icon :
5	Double-click on the Multifunction TeSys U to open its Advantys configuration window.
6	click on the **IO Image** tab

7	Set the **User Defined Label** fields for each item as follows: 
8	Configure **Ready** and **Tripped** bits of the **Status Register** item to visualize them on the SCADA Electric view. 
9	Click on **OK** and close the Advantys application.

10	A dialog opens and proposes to update the symbols linked to this island :  Click on **Yes** to start the update.
11	Unity Pro creates new variables corresponding to the **User Defined Label** defined in steps 4 & 5. The variables created by Unity Pro are summarized in the table below: **Note:** All the variable names begin with the prefix **IO_LS1_** corresponding to the Device name defined during IO Scanning configuration. Unity Pro only generates a variable if a User Defined Label has been defined in Advantys.

Name	Type	Address	Comment
IO_LS1_Mot3Status0	INT	%MW1030	IO_LS1__R1_S13_TeSys U Sc Mu R_Status Register (455)
IO_LS1_I_Mot3Ready	BOOL	%MW1030.0	IO_LS1__R1_S13_TeSys U Sc Mu R_Ready [Status Register (455)]
IO_LS1_I_Mot3Tripped	BOOL	%MW1030.4	IO_LS1__R1_S13_TeSys U Sc Mu R_Tripped [Status Register (455)]
IO_LS1_Mot3Status1	INT	%MW1031	IO_LS1__R1_S13_TeSys U Sc Mu R_I/O Module Status Register (458)
IO_LS1_Mot3Status2	INT	%MW1032	IO_LS1__R1_S13_TeSys U Sc Mu R_Warning Register (461)
IO_LS1_Mot3Status3	INT	%MW1033	IO_LS1__R1_S13_TeSys U Sc Mu R_Mechanical and Power Supply Status Register (457)
IO_LS1_I_Mot3PKW1	INT	%MW1034	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Response Object [PKW Service]
IO_LS1_I_Mot3PKW1_WD2	INT	%MW1035	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Response Object [PKW Service]
IO_LS1_I_Mot3PKW2	INT	%MW1036	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Response Data [PKW Service]
IO_LS1_I_Mot3PKW2_WD2	INT	%MW1037	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Response Data [PKW Service]
IO_LS1_Mot3Ctle0	INT	%MW1318	IO_LS1__R1_S13_TeSys U Sc Mu R_Control of the System (704)
IO_LS1_Mot3Ctle1	INT	%MW1319	IO_LS1__R1_S13_TeSys U Sc Mu R_Control of Communication Module (703)
IO_LS1_Mot3Ctle2	INT	%MW1320	IO_LS1__R1_S13_TeSys U Sc Mu R_Output Control (700)
IO_LS1_O_Mot3PKW1	INT	%MW1321	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Request Object [PKW Service]
IO_LS1_O_Mot3PKW1_WD2	INT	%MW1322	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Request Object [PKW Service]
IO_LS1_O_Mot3PKW2	INT	%MW1323	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Request Data [PKW Service]
IO_LS1_O_Mot3PKW2_WD2	INT	%MW1324	IO_LS1__R1_S13_TeSys U Sc Mu R_PKW Request Data [PKW Service]

The following table summarizes the mapping automatically created in Unity Pro for this Multifunction TeSys U connected on an Advantys STB island extension bus.

	ADVANTYS		UNITY PRO	
	Modbus Register	Item	Located address	Variable
Input data	45422	Status 455	%MW1030	IO_LS1_Mot3Status0
	45423	Status 458	%MW1031	IO_LS1_Mot3Status1
	45424	Status 461	%MW1032	IO_LS1_Mot3Status2
	45425	Status 457	%MW1033	IO_LS1_Mot3Status3
	45426 - 45427	PKW Service	%MW1034-35	IO_LS1_I_Mot3PKW1
	45428 - 45429	PKW Service	%MW1036-37	IO_LS1_I_Mot3PKW2
Output data	40005	Control 704	%MW1318	IO_LS1_Mot3Ctrl0
	40006	Control 703	%MW1319	IO_LS1_Mot3Ctrl1
	40007	Control 700	%MW1320	IO_LS1_Mot3Ctrl2
	45426 - 45427	PKW Service	%MW1321-22	IO_LS1_O_Mot3PKW1
	45428 - 45429	PKW Service	%MW1323-24	IO_LS1_O_Mot3PKW2

4.2.4. ATV 312 configuration

The ATV 312 variable speed drive is integrated in the Advantys system using an STB XBE 2100 (CANopen Extension) Advantys module.

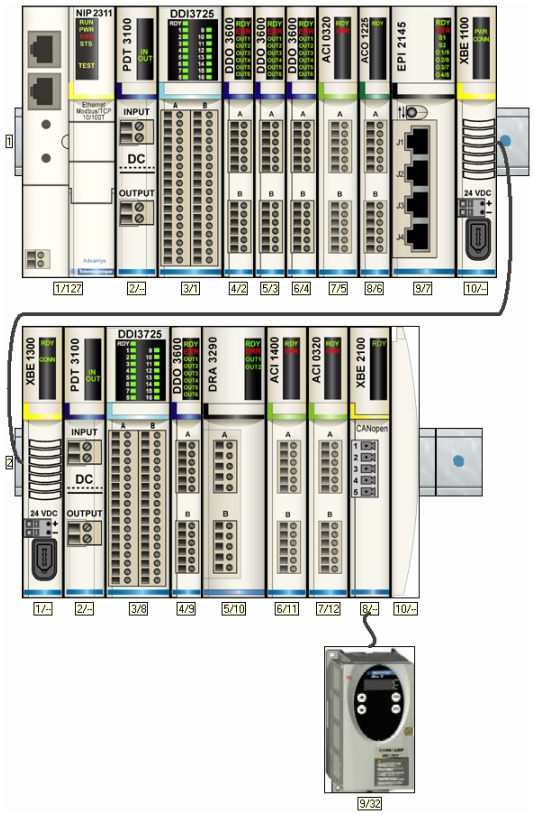
The configuration of motor starter used behind an Advantys STB is performed in two steps:

1) The communication between the Advantys STB island and the PAC is configured in IO Scanning.
2) The data of the selected motor starter is identified inside the Advantys STB Island.

IO Scanning configuration

The Grease & Sand Removal Advantys STB island (172.20.1.21) is used to illustrate the configuration steps.

To configure this island, refer to the previous section describing the Lifting and Screening Advantys STB island as the principle is exactly the same. However, this island is named **IO_GS1_**.



Note: ATV 31 and ATV 312 have not been dissociated within Advantys. While ATV 312 does not exist in the Advantys catalog, the ATV 31 profile is used for the ATV 312 configuration without any differences.

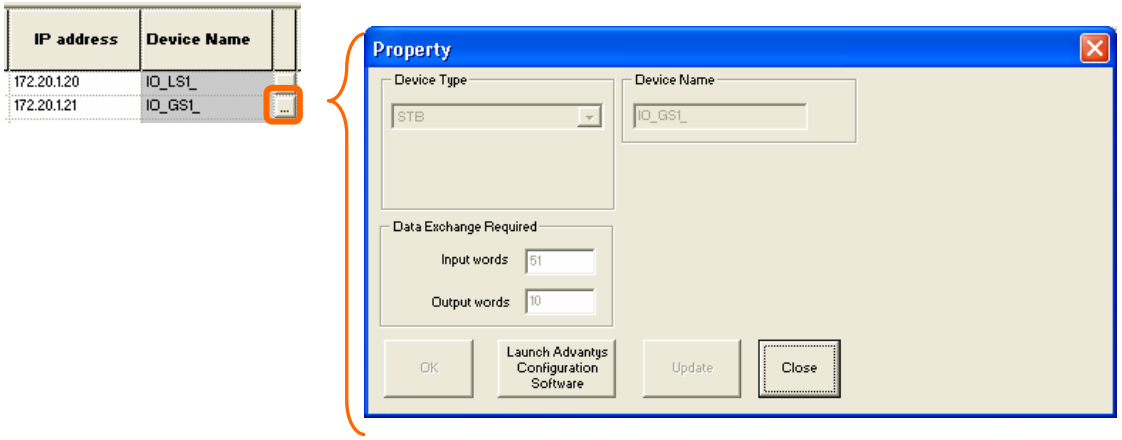
The Grease and Sand Removal Advantys STB island is now configured and ready to communicate with the PAC.

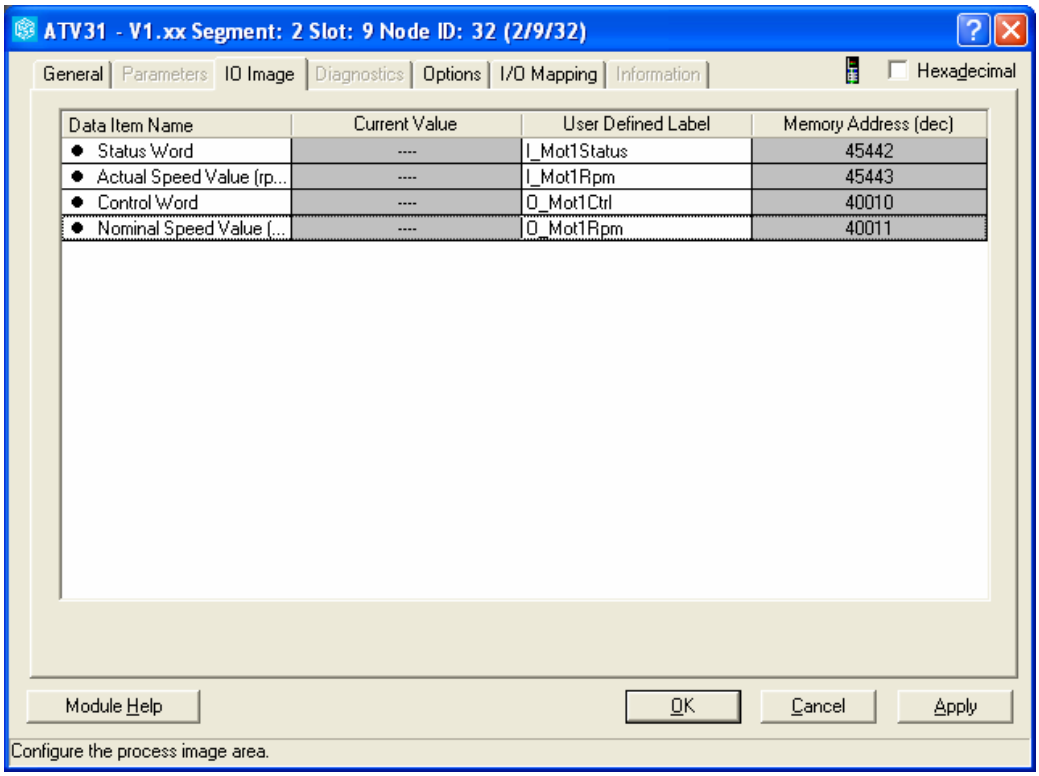
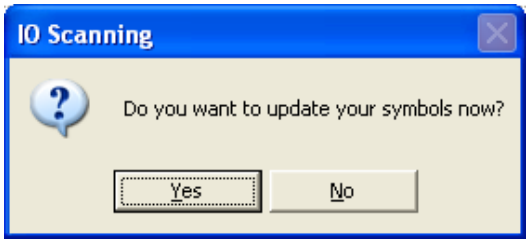
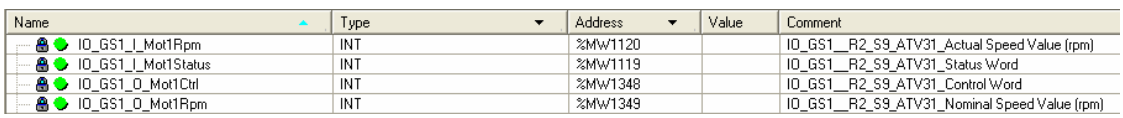
The table below summarizes the mapping created in the IO Scanning for this Advantys STB island.

	ADVANTYS		UNITY PRO		
	Modbus Register	Item	RD Slave index	RD Master object	Length
Input data	45392	First island register input	5391	%MW1070	51
	45442	Last island register input	5441	%MW1120	
Output data	40001	First island register output	0	%MW1340	10
	40010	Last island register output	9	%MW1349	

Define the data of the ATV 312

We use the ATV 312 of the Grease and Sand unit scraper named Mot1 as an example to illustrate the data definition:

Step	Action
1	Open the Grease and Sand unit Advantys STB island configuration to define the data exchanged with the ATV 312.
2	From the IO Scanning service, click on the button in the Device Name **IO_GS1_** field to open the Property window of this device. 
3	Click on the button **Launch Advantys Configuration Software.**
4	To be able to modify the ATV 312 configuration, unlock the island configuration by clicking on the following icon :
5	Double-click on the ATV 31 to open its Advantys configuration window.
6	Click on the **IO Image** tab (rememeber that only the ATV 31 profile is available in Advantys).

7	Set the **User Defined Label** fields for each item as follows: 
8	Click on **OK** and close Advantys.
9	A dialog box opens and proposes to update the symbols linked to this island :  Click on **Yes** to start the update.
10	Unity Pro creates new variables corresponding to the **User Defined Label** defined in step 4. The variables created by Unity Pro are summarized in the table below.  **Note:** All the variable name begins with the prefix **IO_GS1_** corresponding to the Device name configured in IO Scanning. Unity Pro only generates a variable if a User Defined Label has been defined in Advantys.

Name	Type	Address	Value	Comment
IO_GS1_I_Mot1Rpm	INT	%MW1120		IO_GS1__R2_S9_ATV31_Actual Speed Value (rpm)
IO_GS1_I_Mot1Status	INT	%MW1119		IO_GS1__R2_S9_ATV31_Status Word
IO_GS1_O_Mot1Ctrl	INT	%MW1348		IO_GS1__R2_S9_ATV31_Control Word
IO_GS1_O_Mot1Rpm	INT	%MW1349		IO_GS1__R2_S9_ATV31_Nominal Speed Value (rpm)

The table below summarizes the mapping automatically created in Unity Pro for this ATV 312 connected on an Advantys STB island Enhanced CANopen module.

	ADVANTYS		UNITY PRO	
	Modbus Register	Item (CANopen)	Located address	Variable
Input data	45422	Status 6041	%MW1119	IO_GS1_I_Mot1Status
	45423	Speed Value 6044	%MW1120	IO_GS1_I_Mot1Rpm
Output data	40005	Control 6040	%MW1348	IO_GS1_O_Mot1Ctrl
	40006	Speed Setpoint 6042	%MW1349	IO_GS1_O_Mot1Rpm

Note: As for the previous island, the configuration files of this Advantys STB island generated by Advantys application are located in a directory defined in the Unity Pro options. Therefore, to use the Unity Pro application on another computer and make modifications concerning the STB islands mapping, it is necessary to copy the Unity Pro application as well as the Advantys configuration files into the proper directory of the new computer.

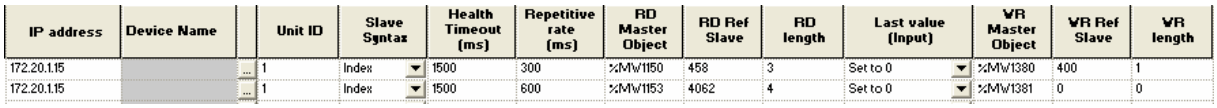
4.2.5. ATS 48 Configuration

The ATS 48 is connected to the Ethernet network through an ETG100 gateway. Communication is transparent between the Ethernet network and Modbus Serial line. Therefore, the IO Scanning service directly accesses the ATS 48 data. The **Unit ID** identifies the slave address of the device on the Modbus Serial line.

In the following example, three status registers and four measure registers are read, and one command register is written.

Because the status and measure registers are not in a continuous area, two IO Scanning lines are required to configure the input data.

Note: All the ATS 48 registers are described in the ATS 48 Modbus user manual.



IP address	Device Name		Unit ID	Slave Syntax	Health Timeout (ms)	Repetitive rate (ms)	RD Master Object	RD Ref Slave	RD length	Last value (Input)	WR Master Object	WR Ref Slave	WR length
172.20.1.15		...	1	Index	1500	300	%MW1150	458	3	Set to 0	%MW1380	400	1
172.20.1.15		...	1	Index	1500	600	%MW1153	4062	4	Set to 0	%MW1381	0	0

Note: The IP address 172.20.1.15 set in the field IP address is the address of the ETG100 Modbus Serial/Ethernet gateway. The value 1 in the Unit ID field corresponds to the Modbus address configured in the ATS 48.

The DDT variables used to communicate with the ATS 48 are presented in the table below. Unlike for the TeSys U, these variables have to be manually created in Unity Pro.

	ATS 48		UNITY PRO	
	Register	Item	Located address	Variable DDT
Input data	458	ETA	%MW1150	IO_LS1_I_PmpR1
	459	ETI	%MW1151	
	460	ETI2	%MW1152	
	4062	LCR	%MW1153	IO_LS1_M_PmpR1
	4063	LTR	%MW1154	
	4064	THR	%MW1155	
	4065	PHE	%MW1156	
Output data	400	CMD	%MW1380	IO_LS1_O_PmpR1

Refer to the ATS 48 documentation for a description of the Modbus registers.

4.2.6. ATS 22 Configuration

As with the ATS 48, the ATS 22 is also connected to the Ethernet network through the ETG100 gateway. Communication is transparent between the Ethernet network and Modbus Serial line. Therefore, the IO Scanning service directly accesses the ATS 22. The **Unit ID** identifies the slave address of the device on Modbus Serial.

In the following example of the ATS 22, 10 status and measure registers are read, and one command register is written.

Unlike the ATS48, the status and measure registers of the ATS 22 are in a contiguous area. Thus, a single line of IO Scanning is necessary.

Note: All the ATS 22 registers are described in the ATS 22 Modbus user manual.

IP address	Device Name		Unit ID	Slave Syntax	Health Timeout (ms)	Repetitive rate (ms)	RD Master Object	RD Ref Slave	RD length	Last value (Input)	WR Master Object	WR Ref Slave	WR length
172.20.1.15		...	2	Index	1500	300	%MW1160	256	10	Set to 0	%MW1390	752	1

Note: The IP address 172.20.1.15 set in the field IP address is the address of the ETG100 Modbus Serial/Ethernet gateway. The value 2 in the Unit ID field corresponds to the Modbus address configured in the ATS 22.

The DDT variables used to communicate with the ATS 22 are presented in the table below. Unlike for the TeSys U, these variables have to be manually created in Unity Pro.

	ATS 22		UNITY PRO	
	Register	Item	Located address	Variable DDT
Input data	256	Status	%MW1160	IO_LS1_I_PmpR2
	257	LCr1	%MW1161	
	258	LCr2	%MW1162	
	259	LCr3	%MW1163	
	260	Voltage	%MW1164	
	261	LI	%MW1165	
	262	Lo	%MW1166	
	263	-	%MW1167	
	264	-	%MW1168	
	265	Freq	%MW1169	
Output data	752	Cmd	%MW1390	IO_LS1_O_PmpR2

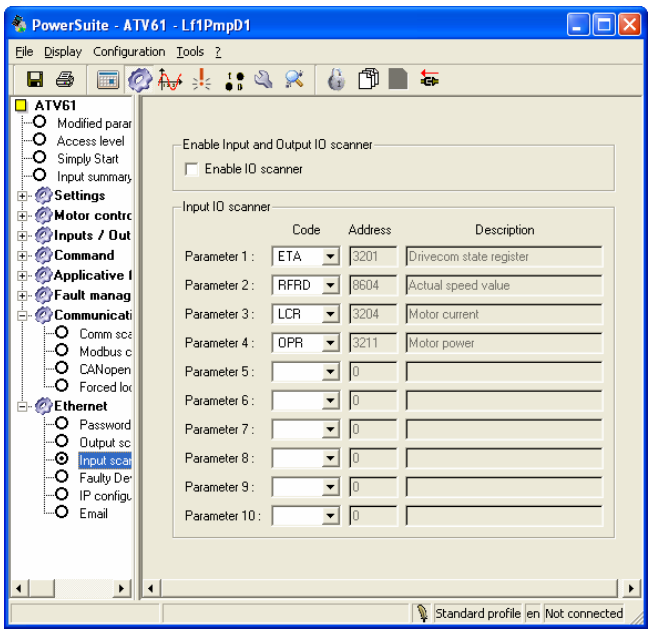
Refer to the ATS 22 documentation for a description of the Modbus registers.

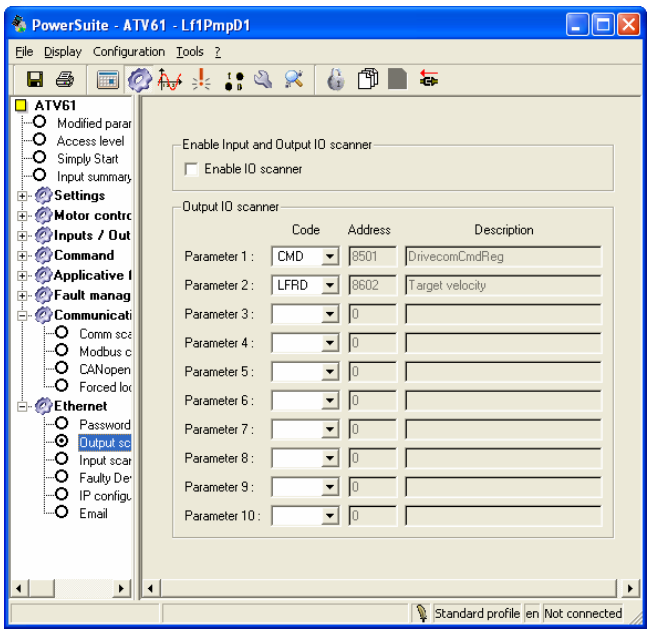
4.2.7. ATV 61 Configuration

Regarding the ATV 61, the input and output IO Scanning parameters can be configured in different ways using:

- PowerSuite software
- ATV 61 graphic display terminal
- ATV 61 Web server

For the ATV 61, the addresses of the registers that are read and written by IO Scanning are defined in the device.
The following example shows the configuration of 4 input and 2 output variables, directly in the device, using PowerSuite software.


PowerSuite - ATV61 - Lf1PmpD1
File Display Configuration Tools ?
ATV61
Enable Input and Output IO scanner
Enable IO scanner
Input IO scanner
Code Address Description
Parameter 1 : ETA 3201 Drivecom state register
Parameter 2 : RFRD 8604 Actual speed value
Parameter 3 : LCR 3204 Motor current
Parameter 4 : OPR 3211 Motor power
Parameter 5 :
Parameter 6 :
Parameter 7 :
Parameter 8 :
Parameter 9 :
Parameter 10 :
Standard profile en Not connected



Then, in Unity Pro, from the **Network configuration** window->**IO Scanning** tab, configure the IO Scanning line as follows:

IP address	Device Name		Unit ID	Slave Syntax	Health Timeout (ms)	Repetitive rate (ms)	RD Master Object	RD Ref Slave	RD length	Last value (Input)	WR Master Object	WR Ref Slave	WR length
172.20.1.50		...	255	Index	1500	50	%MW1130	0	5	Set to 0	%MW1360	0	3

For the ATV 61 IO Scanning configuration, the first register is always reserved:

- Reading length = 1 reserved word + 4 words (configured registers) = 5 words for reading.
- Writing length = 1 reserved word + 2 words (configured registers) = 3 words for writing.

The DDT variables used to communicate with the ATV 61 are presented in the table below. These variables are manually created in Unity Pro.

	ATV 61		UNITY PRO	
	Register	Item	Located address	Variable DDT
Input data	Reserved	-	%MW1130	IO_LS1_I_PmpD1
	3201	ETA	%MW1131	
	8604	RFRD	%MW1132	
	3204	LCR	%MW1133	
	3211	OPR	%MW1134	
Output data	Reserved	-	%MW1360	IO_LS1_O_PmpD1
	8501	CMD	%MW1361	
	8602	LFRD	%MW1632	

Refer to the ATV61 documentation for a description of the different registers.

Note: All the configuration procedures described in this section are also valid for the ATV 71 variable speed drive.

4.2.8. ATV 32 Configuration

The ATV 32 is connected to the Profibus DP network thanks to the PRM, which is a Profibus DP Master used as a gateway between the PAC (on Ethernet) and the ATV 32 Profibus.

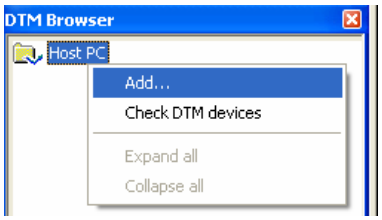
The configuration of the motor starter used behind the PRM requires 3 steps:

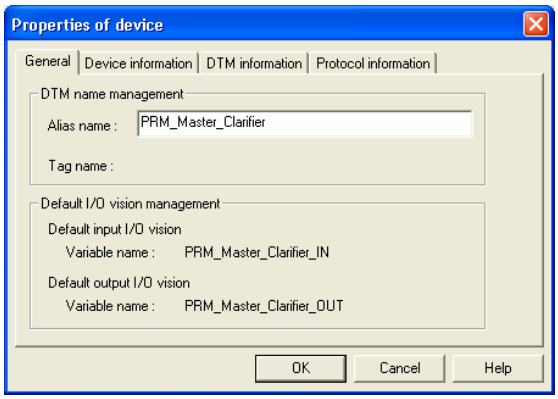
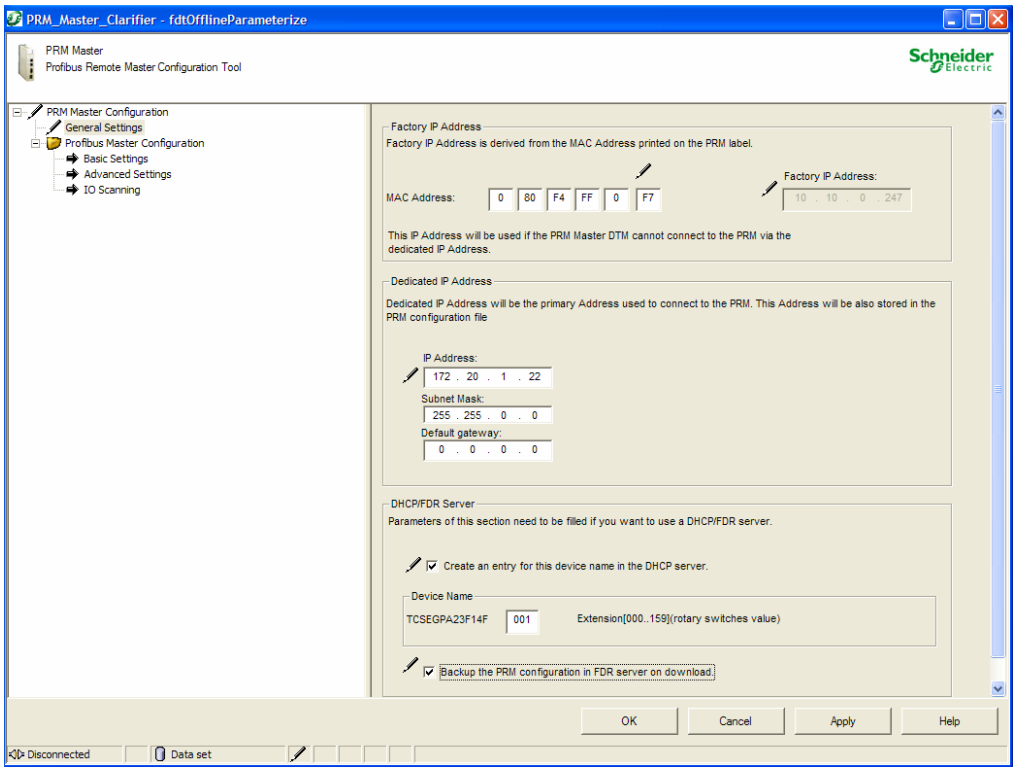
1) declaration and configuration of the PRM in the DTM browser
2) declaration and configuration of the ATV 32 as a Profibus DP slave in the DTM Master
3) Addition of the PRM in the IO Scanning

Note: The configuration of the PRM Master (Ethernet and Profibus DP parameters) and the associated slaves are entirely performed from the DTM Master.

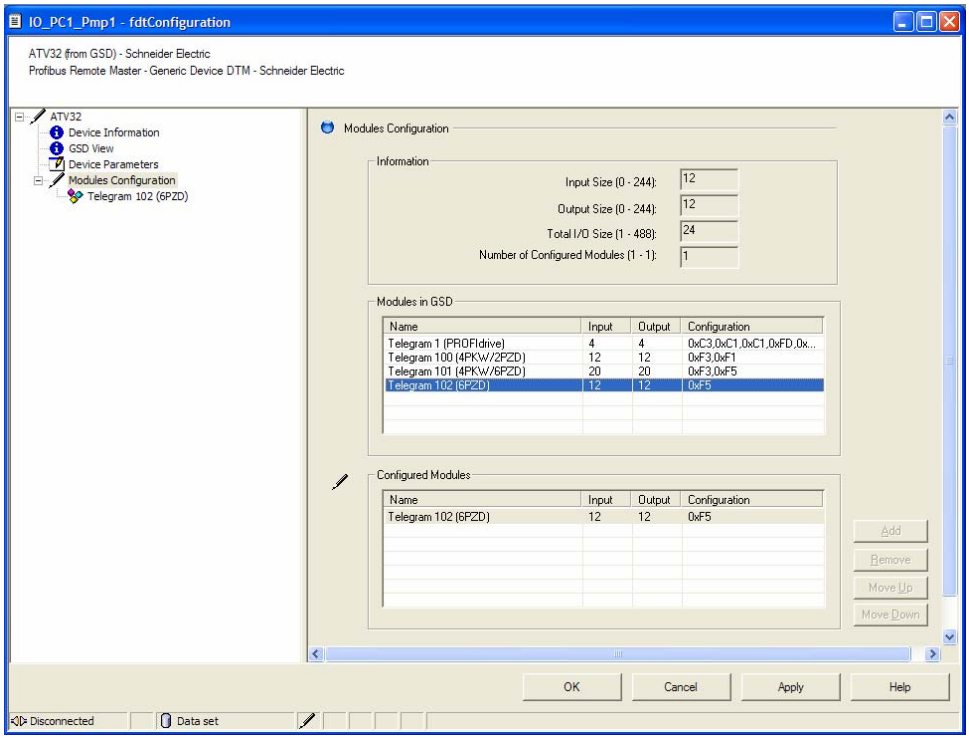
Note: The configuration of the PRM and the ATV 32 require that the PRM Master DTM is installed in the Unity Pro workstation. Then, ensure that the DTM catalog is updated with ATV 32 gsd file and PRM Master DTM. Refer to the PRM documentation for more information.

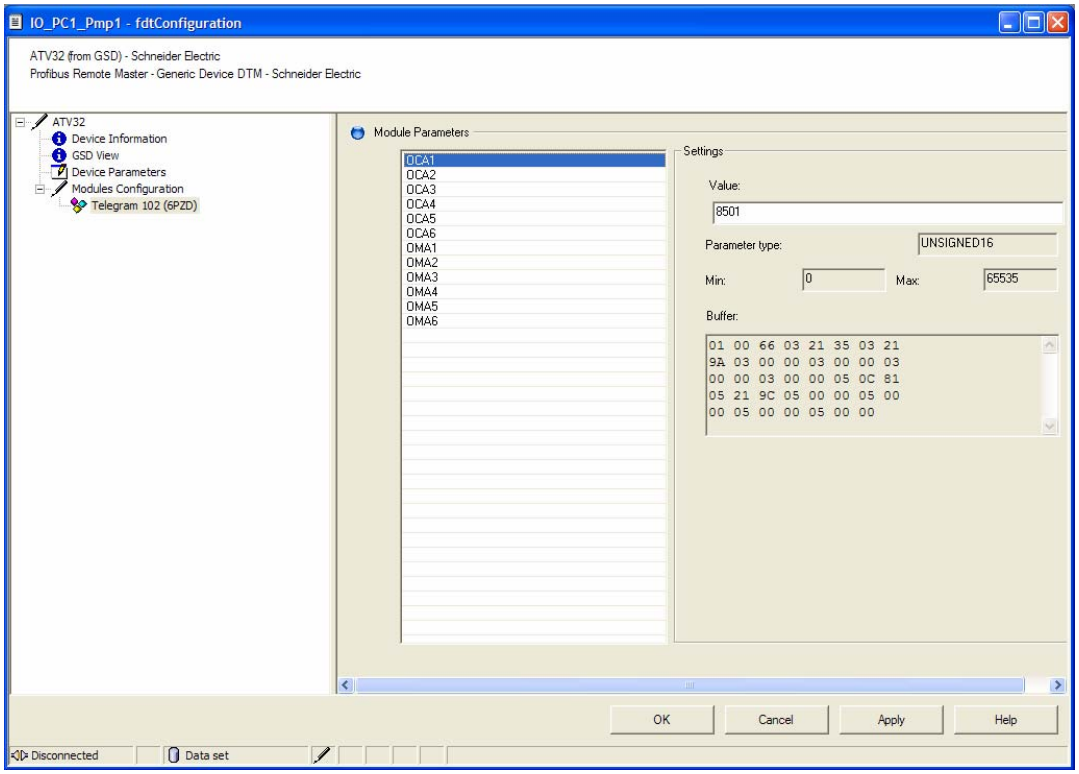
Definition and configuration of the PRM in the DTM Browser

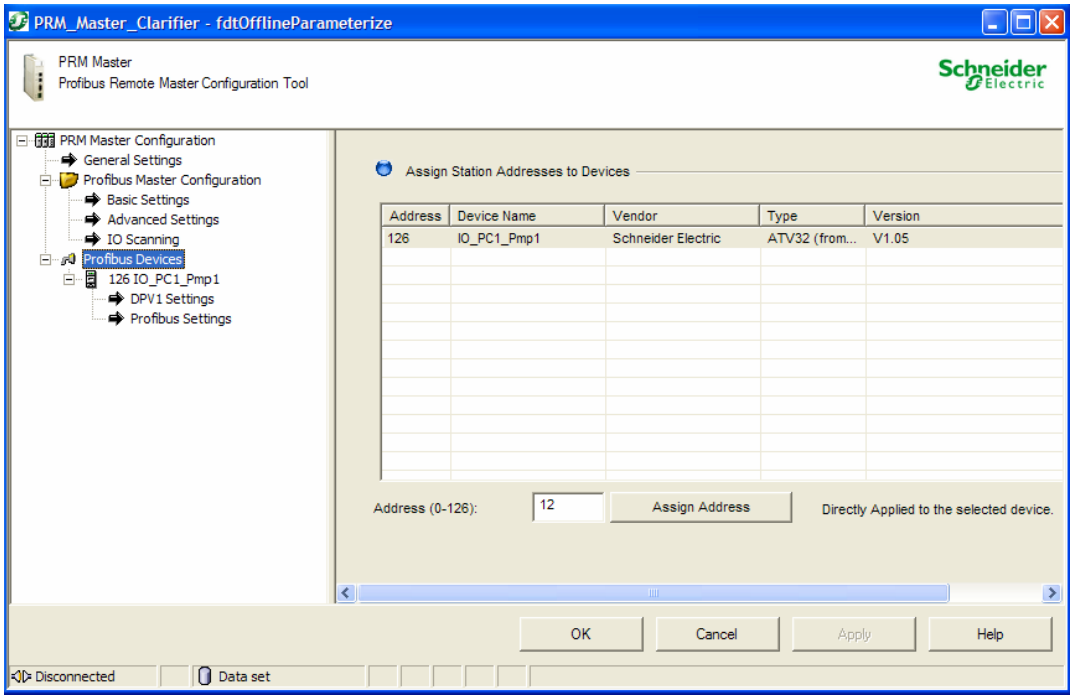
Step	Action
1	Run Unity Pro and open the Water project.
2	From the Tools menu, select **DTM Browser**.
3	From the DTM Browser window, right click on **Host PC** and select **Add…** to open the communication devices catalog. 
4	Select the PRM_Master and click on the button **Add DTM** **Note:** To be able to select the PRM Master, the PRM Master DTM must have been previously installed and the catalog updated and reloaded.
5	**Result:** A confirmation dialog box opens, confirm the modification.

6	A device property window opens. Modify the PRM name, if necessary. In our case, it is named PRM_Master_Clarifier. 
7	Click on **OK** to confirm.
8	From the DTM Browser window, double click on the PRM_Master_Clarifier instance to open its properties. Click on **General Settings** and fill in the PRM **MAC address**, **IP address** and DHCP/FDR parameters as described in the screenshot below:  Profibus baud rate in the Basic Setting menu is set to1500 kbits/s. The others parameters are left with default values.
9	Click on OK to validate the modifications and close the window. The PRM of the Clarifier unit is now declared and configured.

Definition and configuration of the ATV 32 slave device

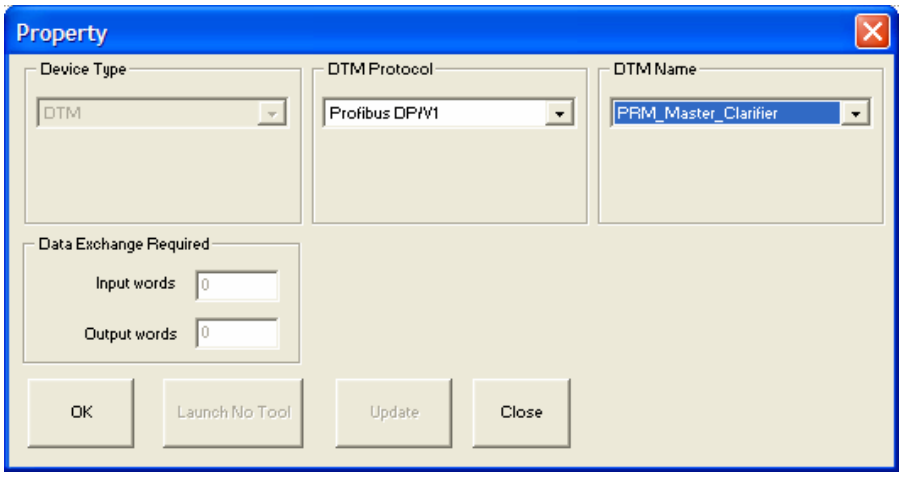
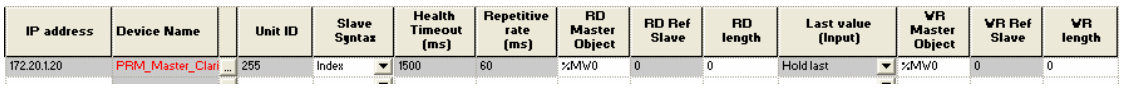
Step	Action
1	From the **DTM Browser** window, right click on the **PRM_Master_Clarifier** device and select **Add…** to open the device catalog.
2	Select the ATV 32 and click on **Add DTM** button. **Note:** The GSD of the devices must have been previously added to the catalog. **Result:** A confirmation dialog box opens.
3	**Confirm** the modification.
4	A device property window opens. Modify the starter name. In our case, it is named **IO_PC1_Pmp1**.
5	Click on **OK** to confirm. **Note:** Once the slave device is added, a Profibus address by default is assigned. This address can be modified in PRM Master DTM (see step 7).
6	Double click on the Profibus slave to open its Profibus configuration window. In the list on the left, select **Modules Configuration**. 4 module types are available to communicate with the ATV 32. Choose **Telegram 102** by clicking on the button **Add**. **Note:** The DFB provided with this guide is configured to be used with Telegram 102. 

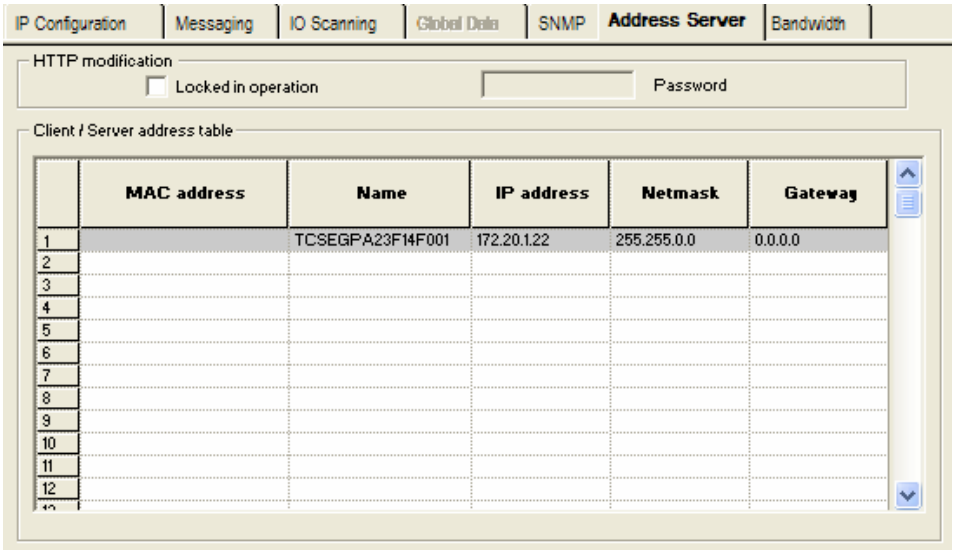
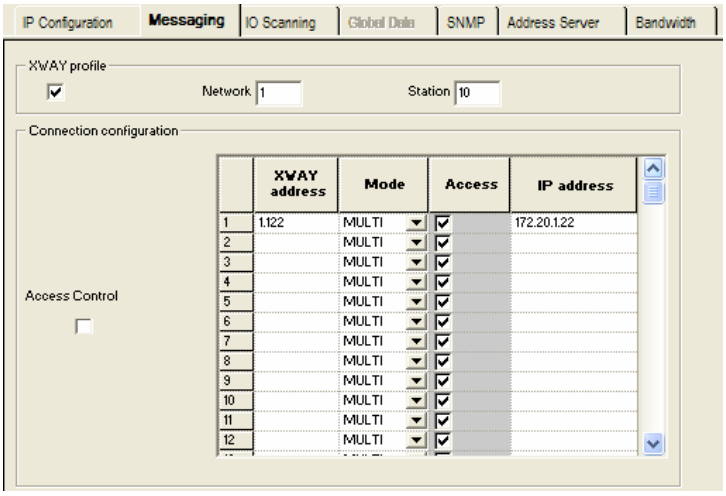
7	Click on **Telegram 102** in the tree list on the left to define the registers that are used on the ATV 32 from the PAC.  In our case, and to be consistent with the DFB MOT_ATV32_Pfb provided with this guide, registers of this Telegram are defined as described below. **Module Parameters** / **Registers** / **Module Parameters** / **Registers** OCA1 8501 OMA1 3201 OCA2 8602 OMA2 8604 OCA3 0 OMA3 3204 OCA4 0 OMA4 3208 OCA5 0 OMA5 0 OCA6 0 OMA6 0 Refer to the ATV32 documentation for a complete description of the register numbers.
8	Click on **OK** to finish the configuration of the Profibus DP slave.

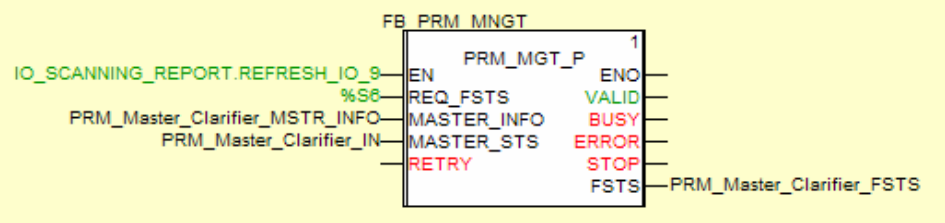
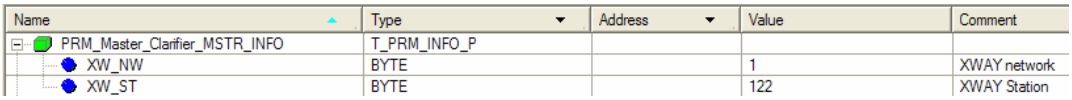
9	To modify the ATV 32 Profibus address, double click on PRM_Master_Clarifier from the **DTM Browser** to open the PRM configuration window. Select **Profibus Devices** in the list on the left as described in the screenshot below: 
10	Select the ATV 32 and assign a new address (12 in our case). **Note:** This address must be identical to the one configured on the ATV 32.
11	Click on **OK** to finish.
12	On the ATV 32 device equipped with a Profibus DP communicator, configure the Profibus DP address as follows: Menu: **CONF** -> **FULL** -> **COM** -> **Cbd** -> **AddrC** and set the address to 12 as defined in the PRM configuration. **Note:** This menu can be used only if the Profibus DP communicator is plugged.

The PRM Master and ATV 32 slave are now configured.

Add the PRM in the IO Scanning

Step	Action
1	We can now start the IO Scanning configuration. From the Ethernet network configuration window, click on **IO Scanning** tab.
2	On a new IO Scanning line, click on the button in the Device Name field to open the Property window. **Note:** The Device Type DTM is automatically selected in the Property window if the new IO Scanning line has been left totally blank. 
3	Select the new PRM **PRM_Master_Clarifier** in the combo box **DTM Name** and click on **OK**. A pre-defined line is then added for this PRM in IO Scanning:  **Note:** The Device Name field is written in red because Unity Pro asks for an update to finish the configuration.
4	On the same line, fill in the start address (Master Object) + word number (length) for the read and write registers: • %MW1170 and 100 words for read register. • %MW1400 and 100 words for write register. **Note:** It is necessary to configure the **RD length** and **WR length** values to start the PRM update but these values are not yet defined. Therefore, the read/write word lengths are overestimated in the first step and are fine-tuned later.
5	**Validate** to accept the IO Scanning settings.

6	Open the Property window (using the button in the Device Name field) and click on the **Update** button.
7	**Result:** the Device Name field appears in black, PRM and I/O variables have been automatically generated and the Address Server has been configured. Click on the **Address Server** tab to check the configuration. 
8	Premium PAC specificities: To run and use a PRM in Premium PAC architecture, it is necessary to declare an XWAY profile. Click on **Messaging** tab.
9	Select an XWAY profile on the PAC Ethernet module and an XWAY connection (1.122 in our case) associated with IP address 172.20.1.22 corresponding to the PRM. **Note:** XWAY station number must be greater than 100. 

10	Click on **Validate** to accept the network settings.
11	To start the PRM, the PRM_MGT block must be called in the application. It activates the I/O exchanges on Profibus. Create a new section called PRM_Management.
12	Instantiate the DFB PRM_MGT_P in this section. There is one DFB for each type of PAC: PRM_MGT_P for Premium, PRM_MGT_M for M340, and PRM_MGT_Q for Quantum. **Note:** The Profibus library provided with the PRM must be installed before implementing this DFB.  To ensure that the PRM is ready to start, the **IO_SCANNING_REPORT.REFRESH_IO_9** IODDT bit, associated with this device, is used to activate the PRM_MGT_P DFB. The MASTER_INFO and MASTER_STS inputs must be filled in to authorize the dialog between the PRM and the PAC.
13	**MASTER_INFO** provides the communication path for the PRM. In the case of a Premium PAC, the XWAY path is specified. This variable has been automatically generated during the PRM update performed in the IO Scanning. Edit this DDT type variable to set the XWAY address configured in the previous step as the default one:  **MASTER_STS:** This input is connected to the **PRM_Master_Clarifier_IN** variable. This variable has been automatically generated during the PRM update performed in the IO Scanning.
14	Perform a Build All of the Unity Pro program and transfer it to the PAC.
15	Finally, transfer the complete configuration in the PRM: From the DTM Browser, right click on device **PRM_Master_Clarifier** and select **Connect**.

16	Once connected, right click again on the PRM and select **Store data to device** to upload the configuration in the PRM. At the end of the transfer, the PRM disconnects and restarts.

The PRM of the Clarifier Unit is now configured and communicates with the PAC.

4.3. SCADA System Configuration

Without going into details about the Vijeo Citect programming, this section outlines the principle stages. The configuration of OFS, which is used for the communication between the SCADA system and the PAC, is also described.

4.3.1. Principle Stages

- Creating new Cluster

From Citect **Project Editor**->**Servers**->**Clusters**, create Cluster1

- Creating new Network address

From Citect **Project Editor**->**Servers**->**Network Addresses**, enter the computer name or IP address of the OFS server

- Creating new Alarm Server

From Citect **Project Editor**->**Servers**->**Alarm Servers**, create Alarm Server called MyAlarms linked to Cluster1

- Creating new Trend Server

From Citect **Project Editor**->**Servers**->**Trend Servers**, create Trend Server called MyTrend linked to Cluster1

- Creating new I/O Server

From Citect **Project Editor**->**Servers**->**IO Servers**, create I/O Server called OPC linked to Cluster1

- Creating new I/O Device

Create a Premium I/O Device by using the communication wizard and link it to the I/O Server OPC previously created, and select OFS communication protocol.

The SCADA I/O tag database is created in Vijeo Citect from the Unity Pro variable database and thanks to OFS, Vijeo Citect is able to communicate with unlocated PAC variables.

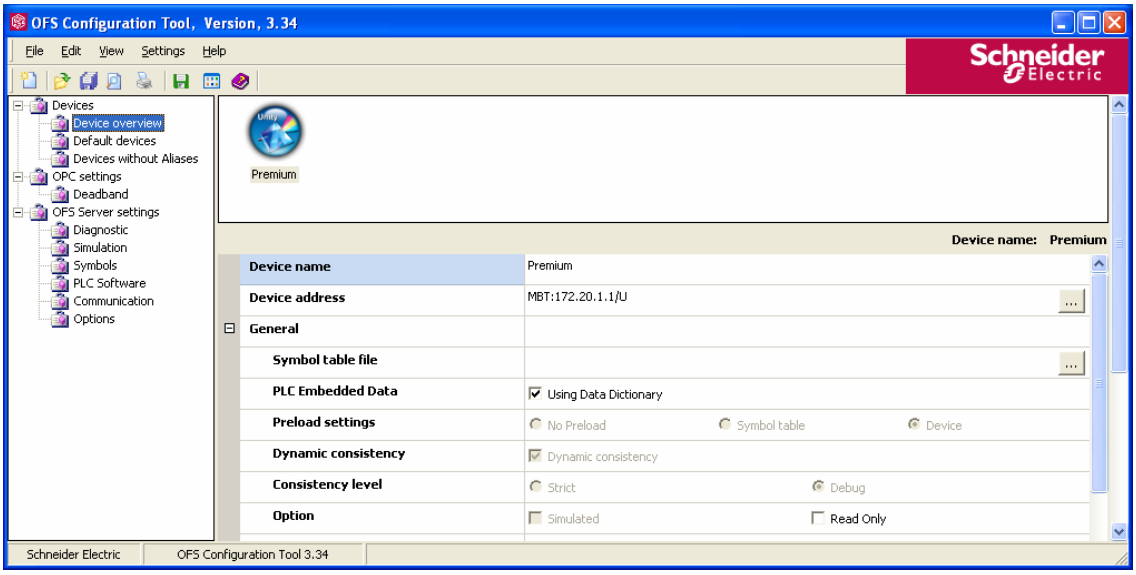
4.3.2. OFS Configuration

From OFS configuration tool

Create an Alias whose type is **Unity Pro** and name it, for example, “Premium”. Set the IP address of the PAC in the corresponding field.

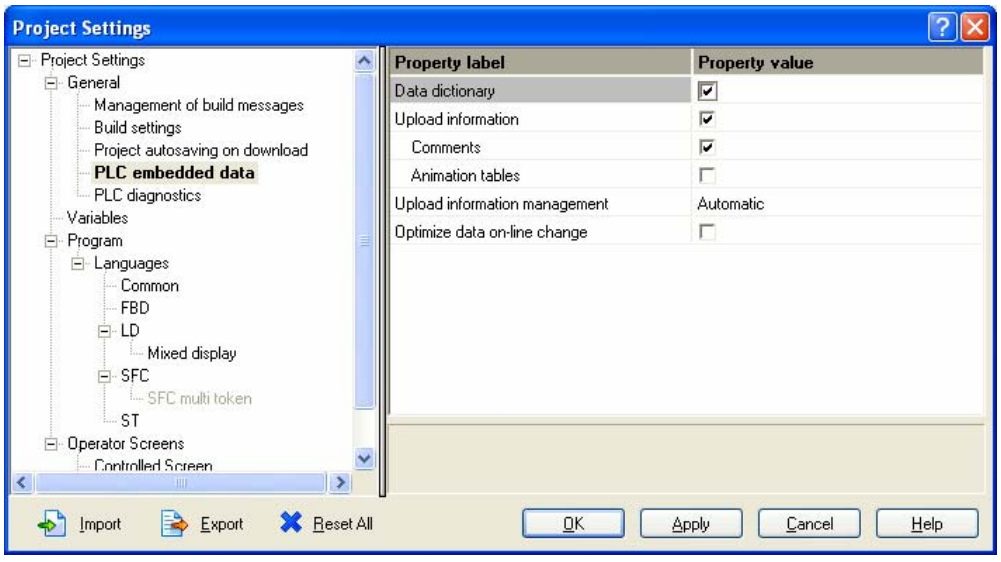
Select **Using Data Dictionary**.

This option allows OFS to connect directly to the PAC to get the embedded database and be regularly updated.

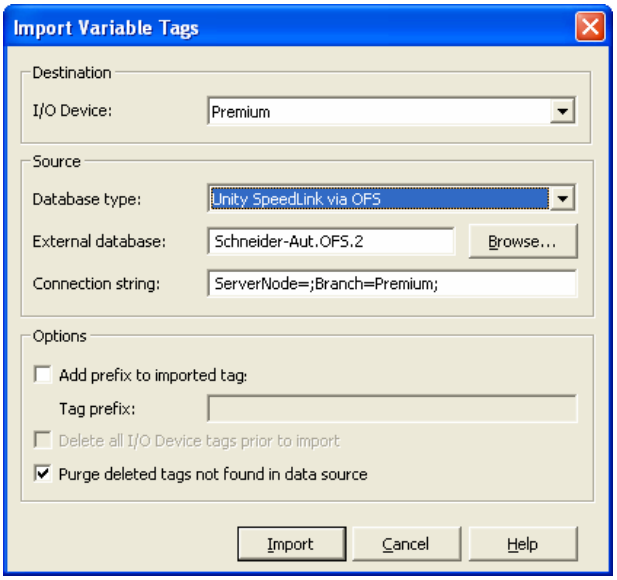
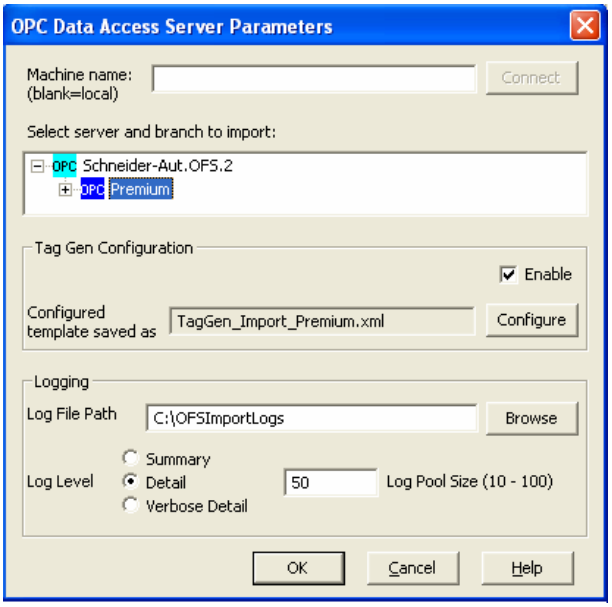


From Unity Pro

In the Unity Pro project settings, the Data dictionary option has to be selected in order to allow OFS to read the embedded database.



4.3.3. Importing tags in Vijeo Citect

Step	Action
1	In Vijeo Citect, from **Citect Explorer** menu, click on **Tool** -> **Import Tags**. Select the **Premium** I/O Device, and then select the **Unity SpeedLink via OFS** database type.
2	 Click the **Browse** button to connect to the OFS server.
3	 Select the displayed OPC group Premium corresponding to the Alias previously created under the OFS server and click on **OK.**
5	Back in the first window, select **Purge deleted tag not found in data source** to avoid variables list overload.
6	Finally, click the **Import** button to start the import process.

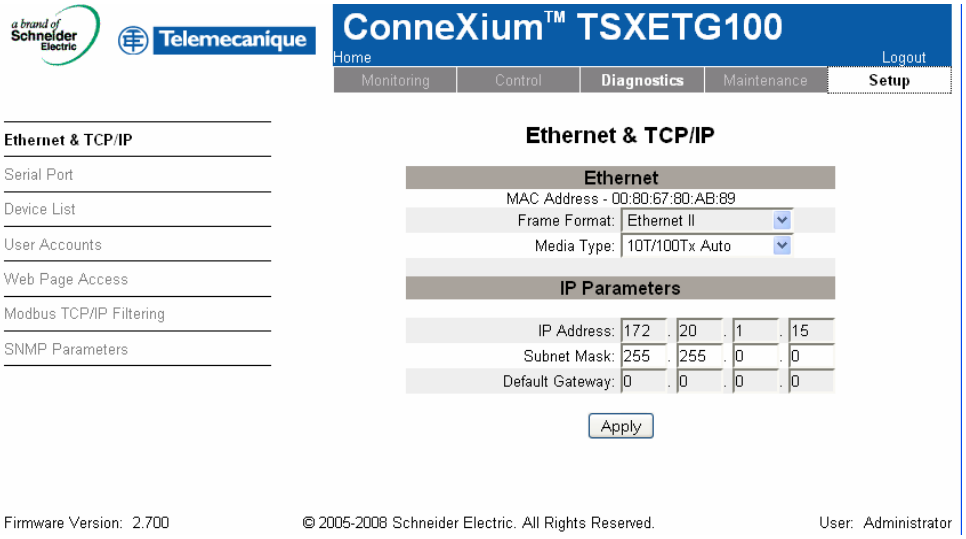
4.4. ETG100 Configuration

4.4.1. Connection to the Gateway

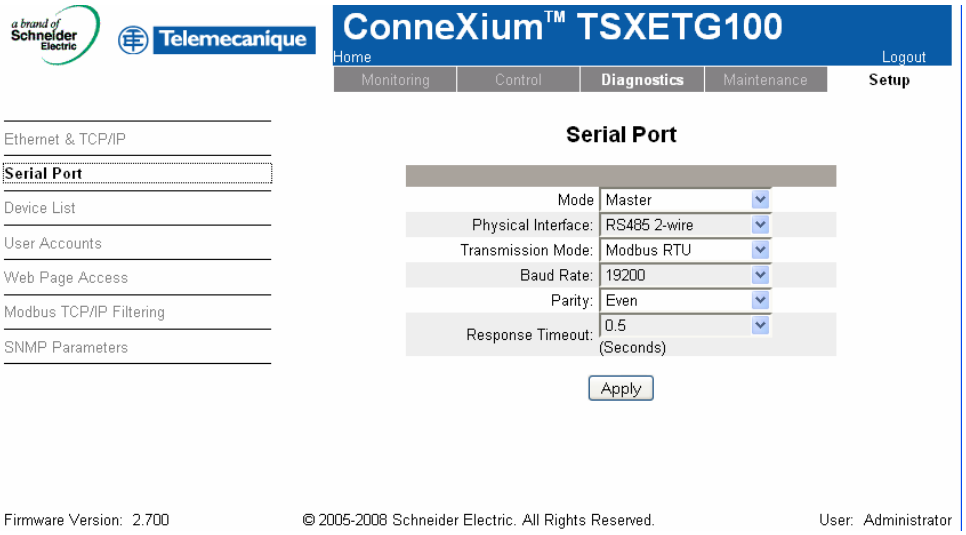
The ATS 48 and ATS 22 are connected to the Ethernet network via an ETG100 gateway. The communication becomes transparent between the Ethernet network and Modbus serial line. Thus, the IO Scanning service can directly access the data of the devices.

To configure this gateway for the first time, use a web browser and log on to the default IP Address. Use login **Administrator** and password **Gateway** to log in.

Ethernet Parameters configuration



Modbus Parameters configuration



Note: The I/O Scanner sends several requests in parallel while the gateway sends them one by one on the Modbus serial line.

The time delay generated by this communication mode can trigger a gateway timeout if a Modbus slave communication is inoperative. An I/O scanner timeout can be triggered for all the slaves handled by the gateway.

To minimize this, set a minimum time-out (0.5 s) on the serial port of the gateway.

5. Implementation

The main purpose of this chapter is to describe the implementation of the components introduced in the Design chapter.

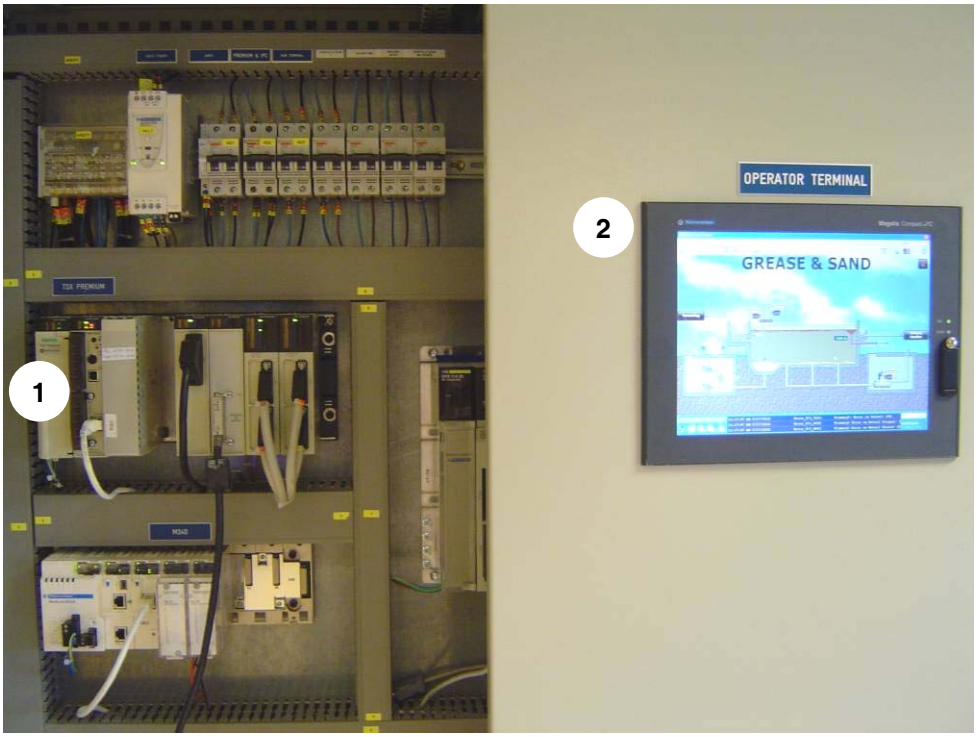
5.1. Hardware Implementation

The different process steps of the pre-treatment part of a wastewater station are split into three cabinets:

- Main control cabinet
- Lifting and Screening units cabinet
- Grease & Sand Removal and Primary Clarifier units cabinet

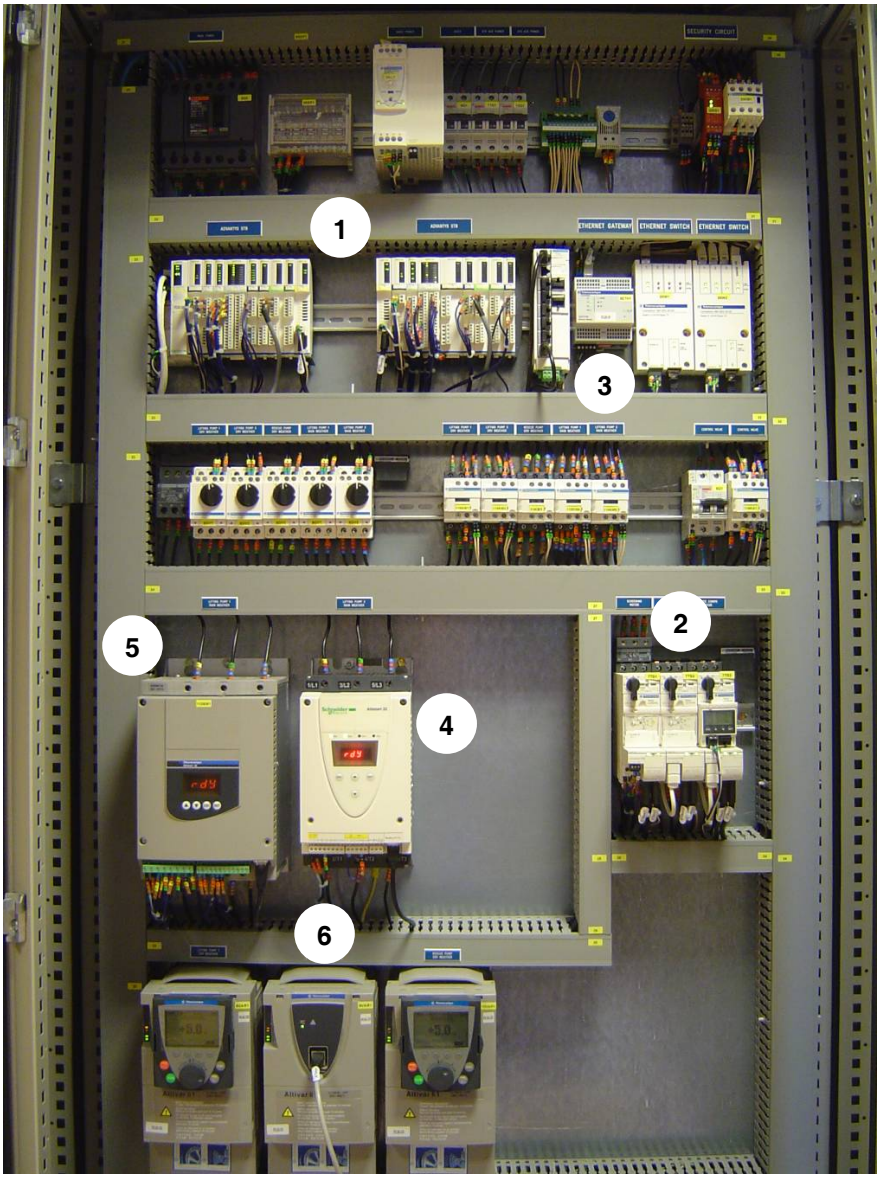
5.1.1. Main control cabinet

The main control cabinet contains the Premium PAC (1) (CPU TSX P57 4634) with an integrated Ethernet module. This PAC manages the distributed I/Os and motor control devices located in other cabinets. The standalone SCADA (Vijeo Citect) application connected to the PAC is run on a Magelis iPC computer (2). This application monitors and controls the global process.



5.1.2. Lifting and Screening units cabinet

This cabinet integrates the Lifting and Screening hardware.



1: Advantys STB islands on the Ethernet network

2: TeSys U Standard, Advanced and Multifunction on the Advantys extension bus

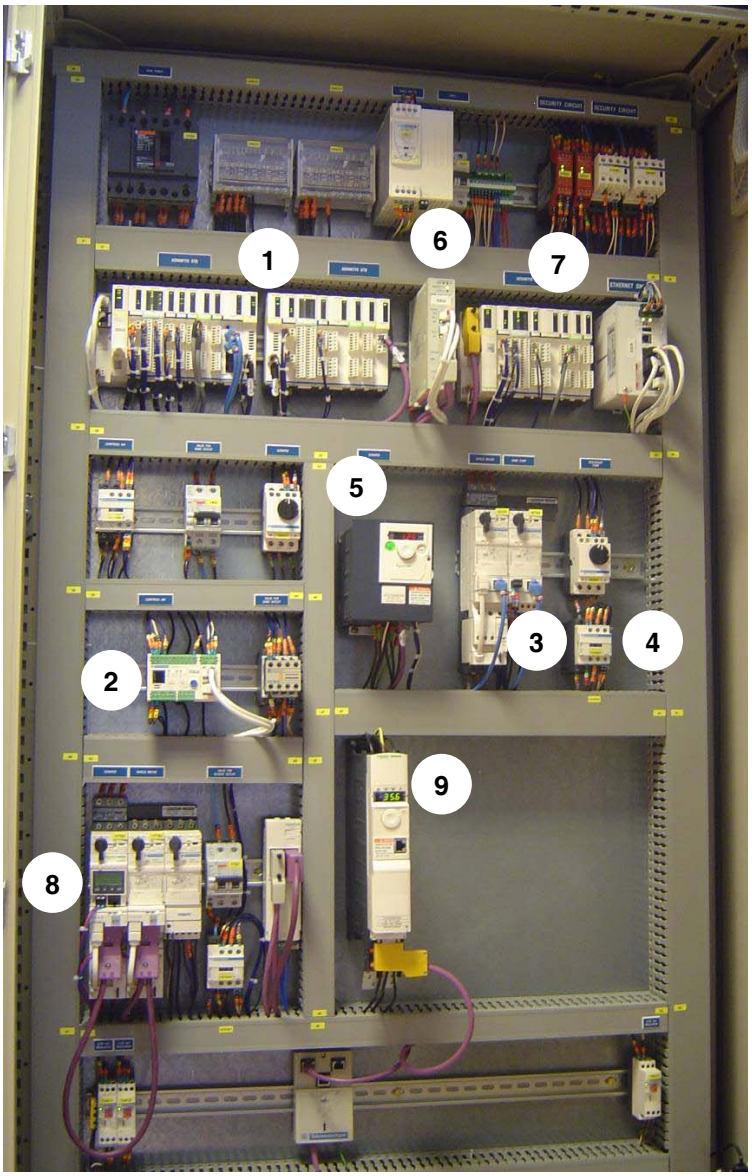
3: ETG100 Modbus serial – Ethernet gateway

4: ATS 22 Soft Starter on the Modbus serial line (controlled through the ETG100 gateway)

5: ATS 48 Soft Starter on the Modbus serial line (controlled through the ETG100 gateway)

6: ATV 61 Variable speed drives on the Ethernet network

5.1.3. Grease & Sand Removal and Primary Clarifier units cabinet



1: Advantys STB (dual port) island on the Ethernet network daisy chain loop

2: TeSys T on the Ethernet network daisy chain loop

3: TeSys U Standard on pre-wired Advantys STB interface EPI2145 module

4: TeSys D contactor wired on Advantys STB I/O module

5: ATV 312 on CANopen using an STB XBE 2100 (CANopen Extension) Advantys STB module

6: PRM

7: Advantys STB island on the Profibus DP network (controlled through the PRM gateway)

8: TeSys U Standard and Multifunction on the Profibus DP network (controlled through the PRM gateway)

9: ATV 32 variable speed drive on the Profibus DP network (controlled through the PRM gateway)

5.2. DFB Implementation in the PAC

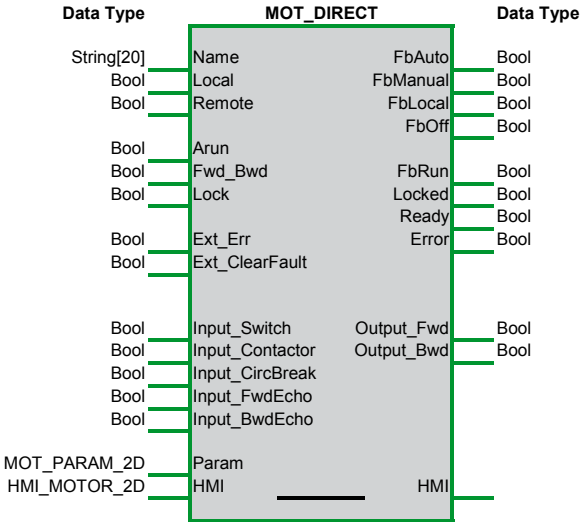
Each type of starter is associated with 1 dedicated DFB. The table below summarizes the DFBs that we created for our application.

STARTER	Connection Type	DFB name	IO SCANNING DATA	HMI STRUCTURE	PARAM STRUCTURE
ATV 61	Modbus/TCP	MOT_ATV61_71	Input : ATV61_71_IOSCAN_I Output : ATV61_71_IOSCAN_O	HMI_MOTOR_A	MOT_PARAM_SPEED
ATV 71					
ATV 32	Profibus DP	MOT_ATV32_Pfb	Input 1 : ARRAY[0..3] OF BYTE Input 2 : ARRAY[0..3] OF BYTE Output : ARRAY[0..3] OF BYTE	HMI_MOTOR_E	
ATV 31	Advantys CANopen	MOT_ATV31_312	Input : 2 INT Output : 2 INT	HMI_MOTOR_C	
ATV 312					
ATS 48	Modbus via ETG100	MOT_ATS48	Input 1 : ATS48_IOSCAN_I Input 2 : ATS48_IOSCAN_IM Output : ATS48_IOSCAN_O	HMI_MOTOR_B	MOT_PARAM
ATS 22		MOT_ATS22	Input : ATS22_IOSCAN_I Output : ATS22_IOSCAN_O	HMI_MOTOR_D	
TeSys T	Modbus/TCP	MOT_TESYST_Eth	Input : ARRAY[0..3] OF INT Output : 1 INT	HMI_MOTOR_TT	MOT_PARAM_2D
TeSys U Std	Modbus/TCP via Advantys extension rack	MOT_TESYSU_s	Input : 2 INT Output : 3 INT	HMI_MOTOR_TU_s	
	Profibus DP	MOT_TESYSU_s_Pfb	Input : ARRAY[0..7] OF BYTE Output : ARRAY[0..3] OF BYTE	HMI_MOTOR_TU_DP_2D	
TeSys U Adv.	Modbus/TCP via Advantys extension rack	MOT_TESYSU_a	Input : 3 INT Output : 3 INT	HMI_MOTOR_TU_a	
	Profibus DP	MOT_TESYSU_a_Pfb	Input : ARRAY[0..7] OF BYTE Output : ARRAY[0..3] OF BYTE	HMI_MOTOR_TU_DP_2D	
TeSys U Mult.	Modbus/TCP via Advantys extension rack	MOT_TESYSU_m	Input : 4 INT Output : 3 INT	HMI_MOTOR_TU_m	
	Profibus DP	MOT_TESYSU_m_Pfb	Input : ARRAY[0..7] OF BYTE Output : ARRAY[0..3] OF BYTE	HMI_MOTOR_TU_DP_2D	
TeSys U Std	Modbus TCP via Advantys STB + Prewiring	MOT_DIRECT	Input : 6 BOOL Output : 2 BOOL	HMI_MOTOR_2D	
TeSys D	Modbus TCP via Advantys STB I/O Modules				

All the DFBs described here are provided with this document and are not locked. As a result, it is possible to open and analyze them using Unity Pro.

5.2.1. TeSys D Controller

This section describes the MOT_DIRECT DFB. It controls a TeSys D contactor and a TeSys U Standard connected on a pre-wired Advantys interface EPI2145 module.



The communication with the device is performed with 6 digital inputs and 2 digital outputs directly accessed from the I/O modules of an Advantys STB via IO Scanning.

	DFB Pin	Type	Description
Input	Input_Switch	BOOL	Switch Status
	Input_Contactor	BOOL	Contactor Status
	Input_CircBreak	BOOL	Circuit Breaker Status
	Input_FwdEcho	BOOL	Forward Feedback
	Input_BwdEcho	BOOL	Backward Feedback
Output	Output_Fwd	BOOL	Command in Forward direction
	Output_Bwd	BOOL	Command in Backward direction

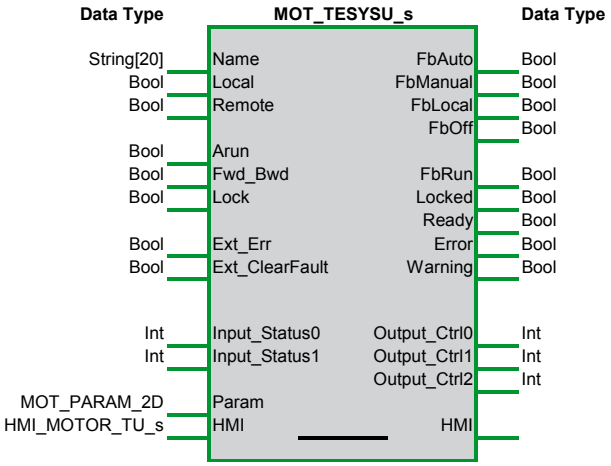
5.2.2. TeSys U controller

This section describes the DFBs for 3 instances of TeSys U (Standard, Advanced and Multifunction) used with:

- Advantys extension bus communication modules
- Profibus DP communication modules

TeSys U Standard

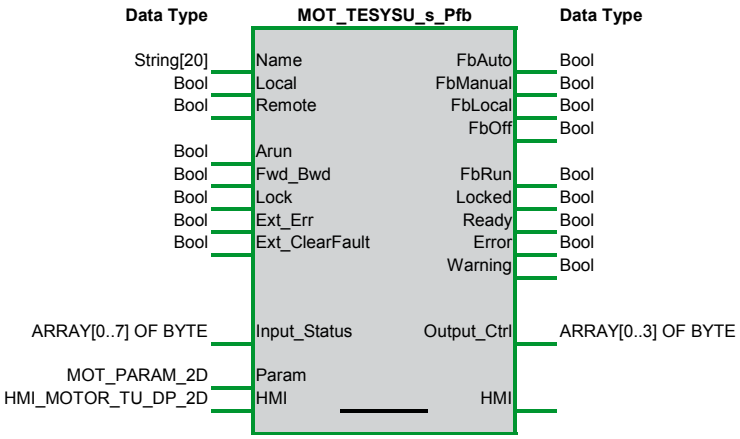
TeSys U Standard on Advantys extension bus



The communication with this device uses 2 input words and 3 output words. These words correspond to the variables automatically generated during the configuration performed with Advantys associated with Unity Pro. Refer to chapter 4.2.3 on Page 65 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status0	455	INT	Switch Status
	Input_Status1	458	INT	Contactor Status
Output	Output_Ctrl0	704	INT	Control of the system
	Output_Ctrl1	703	INT	Control of communication module
	Output_Ctrl2	700	INT	Output Control

TeSys U Standard on Profibus DP

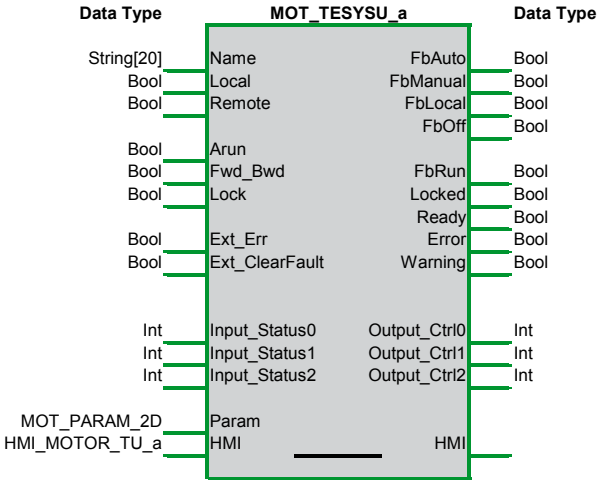


The communication with this device uses 8 input bytes and 3 output bytes. These bytes correspond to the variables automatically generated during the PRM configuration performed with Unity Pro. Refer to chapter 4.2.3 on Page 65 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status	455 LSB	BYTE	Switch Status
		455 MSB	BYTE	Switch Status
		457 LSB	BYTE	Mechanical and power status
		457 MSB	BYTE	Mechanical and power status
		458 LSB	BYTE	I/O Module Status
		458 MSB	BYTE	I/O Module Status
		Reserved	BYTE	-
		Reserved	BYTE	-
Output	Output_Ctrl	704 LSB	BYTE	Control of the system
		704 MSB	BYTE	Control of the system
		700 LSB	BYTE	Output Control
		700 MSB	BYTE	Output Control

TeSys U Advanced

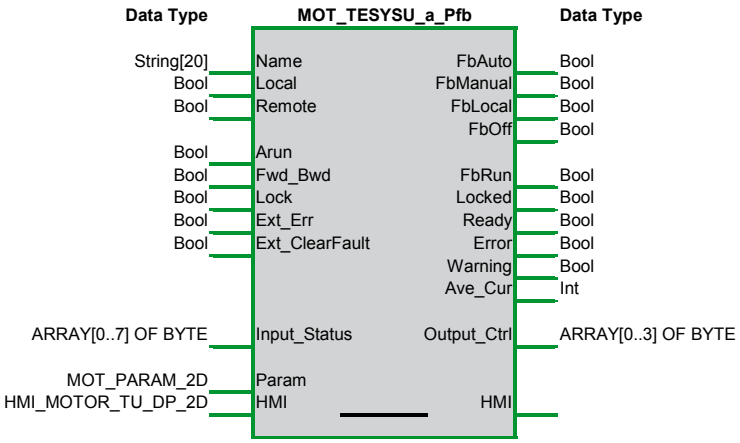
TeSys U Advanced on Advantys extension bus



The communication with this device uses 3 input words and 3 output words. These words correspond to the variables automatically generated during the configuration performed with Advantys associated with Unity Pro. Refer to chapter 4.2.3 on Page 65 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status0	455	INT	Switch Status
	Input_Status1	458	INT	I/O Module Status
	Input_Status2	461	INT	Warning Status
Output	Output_Ctrl0	704	INT	Control of the system
	Output_Ctrl1	703	INT	Control of communication module
	Output_Ctrl2	700	INT	Output Control

TeSys U Advanced on Profibus DP

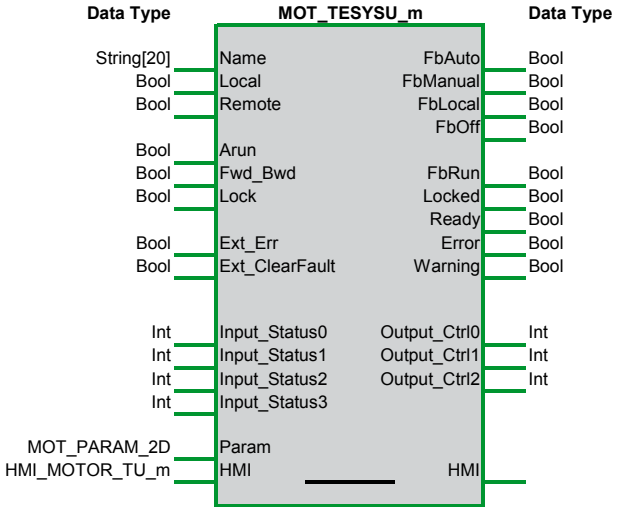


The communication with this device uses 8 input bytes and 4 output bytes. These bytes correspond to the variables automatically generated during the PRM configuration performed with Unity Pro. Refer to chapter 4.2.3 on Page 65 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status	455 LSB	BYTE	Switch Status
		455 MSB	BYTE	Switch Status
		457 LSB	BYTE	Mechanical and power status
		457 MSB	BYTE	Mechanical and power status
		458 LSB	BYTE	I/O Module Status
		458 MSB	BYTE	I/O Module Status
		Reserved	BYTE	-
		Reserved	BYTE	-
Output	Output_Ctrl	704 LSB	BYTE	Control of the system
		704 MSB	BYTE	Control of the system
		700 LSB	BYTE	Output Control
		700 MSB	BYTE	Output Control

TeSys U Multifunction

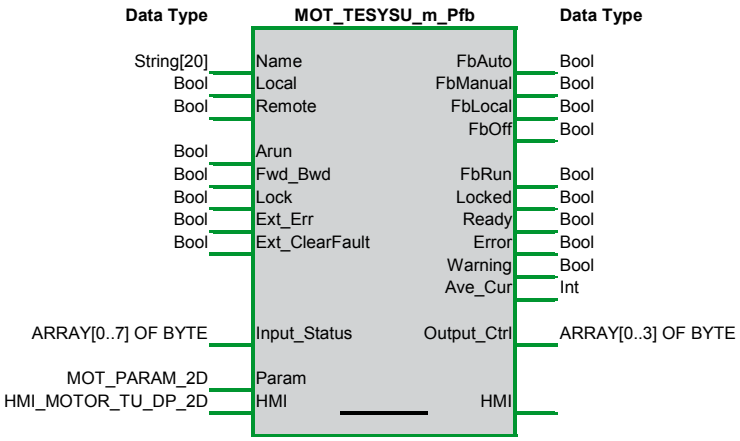
TeSys U Multifunction on Advantys extension bus



The communication with this device uses 4 input words and 3 output words. These words correspond to the variables automatically generated during the configuration performed with Advantys associated with Unity Pro. Refer to chapter 4.2.3 on Page 65 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status0	455	INT	Switch Status
	Input_Status1	458	INT	I/O Module Status
	Input_Status2	461	INT	Warning Status
	Input_Status3	457	INT	Mechanical and power status
Output	Output_Ctrl0	704	INT	Control of the system
	Output_Ctrl1	703	INT	Control of communication module
	Output_Ctrl2	700	INT	Output Control

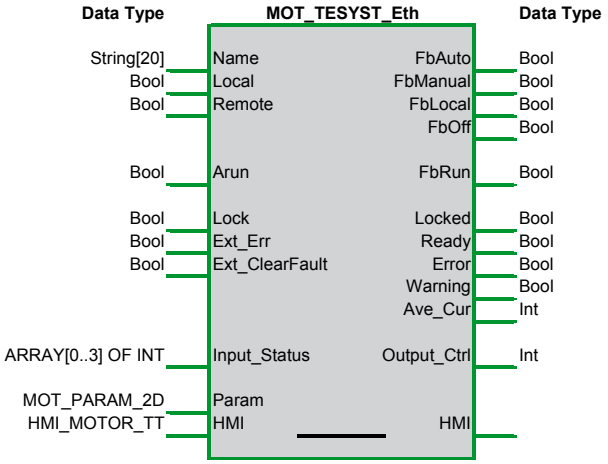
TeSys U Multifunction on Profibus DP



The communication with this device uses 8 input bytes and 4 output bytes. These bytes correspond to the variables automatically generated during the PRM configuration performed with Unity Pro. Refer to chapter 4.2.3 on Page 65 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status	256	INT	Status Register
		257	INT	Phase 1 current
		258	INT	Phase 2 current
		259	INT	Phase 3 current
		260	INT	Voltage
		261	INT	Logic Inputs status
		262	INT	Logic output relays status
		263	INT	-
		264	INT	-
		265	INT	Frequency
Output	Output_Ctrl	752	INT	Control Register

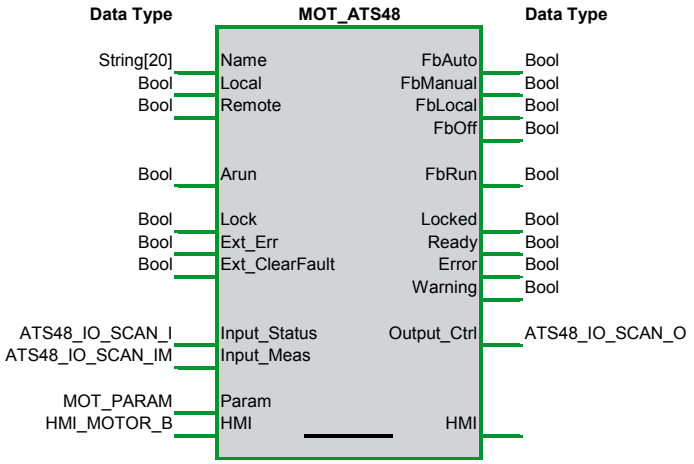
5.2.3. TeSys T Controller on Ethernet



The communication with this device uses 4 input words and 1 output words. They are directly accessed via IO Scanning.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status	2502	INT	Mirror System Status Register 1
		2503	INT	Mirror System Status Register 2
		2504	INT	Mirror Logic Inputs Status
		2505	INT	Logic Ouput Status
Output	Output_Ctrl	2507	INT	Control Register

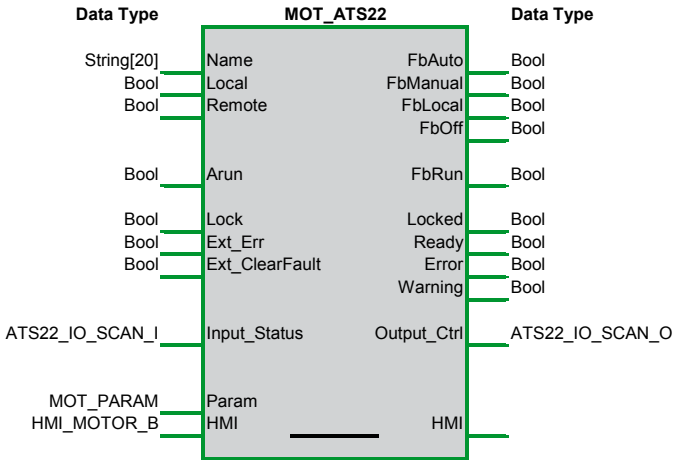
5.2.4. ATS 48 Soft Starter on Modbus



The communication with this device uses 7 input words (3 status words + 4 measure words) and 1 output word for the command, directly accessed via IO Scanning through the TSX ETG100 Gateway. Refer to chapter 4.2.5 on Page 76 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status	458	INT	Status Register
		459	INT	Status Register Extended
		460	INT	Status Register Extended
	Input_Meas	4062	INT	Motot current
		4063	INT	Motor torque
		4064	INT	Motor thermal state
		4065	INT	Phase rotation direction
Output	Output_Ctrl	400	INT	Control Register

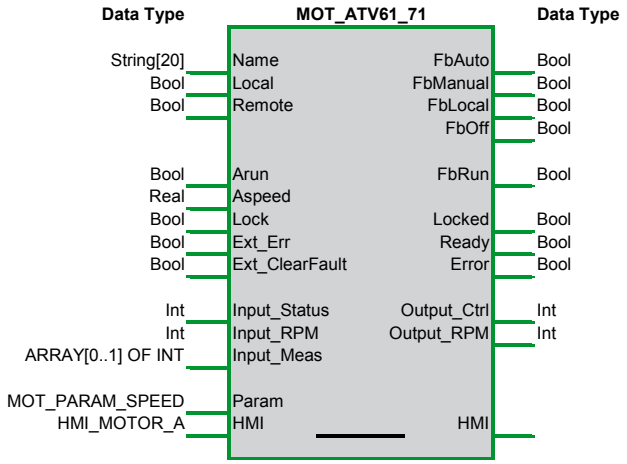
5.2.5. ATS 22 Soft Starter on Modbus



The communication with the ATS 22 uses 10 input words and 1 output word for the command. They are directly accessed via IO Scanning through the TSX ETG100 Gateway. Refer to chapter 4.2.6 on Page 77 for more details.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status	256	INT	Status Register
		257	INT	Phase 1 current
		258	INT	Phase 2 current
		259	INT	Phase 3 current
		260	INT	Voltage
		261	INT	Logical Inputs status
		262	INT	Logic output relays status
		263	INT	-
		264	INT	-
		265	INT	Frequency
Output	Output_Ctrl	752	INT	Control Register

5.2.6. ATV 61 / ATV 71 Variable Speed Drives on Ethernet

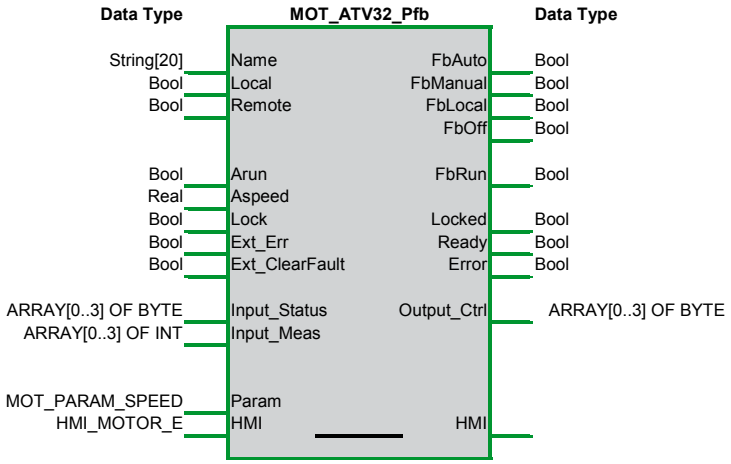


The communication with the ATV 61 and ATV 71 use 5 input words (1 reserved) and 3 output words (1 reserved) for the command. They are directly accessed via IO Scanning. Refer to chapter 4.2.7 on Page 78 for more details.

Note: As this DFB controls a variable speed drive, an input pin named ASpeed has been added so the PAC can specify a speed setpoint for the drive when it runs in automatic mode.

	DFB Pin	Internal Register	Type	Description
Input		Reserved	INT	-
	Input_Status	3201	INT	Status Register
	Input_RPM	8604	INT	Actual speed
	Input_Meas	3204	INT	Motor Current
		3211	INT	Motor Power
Output		Reserved	INT	-
	Output_Ctrl	8501	INT	Control Register
	Output_RPM	8602	INT	Speed setpoint

5.2.7. ATV 32 Variable Speed Drive on Profibus DP



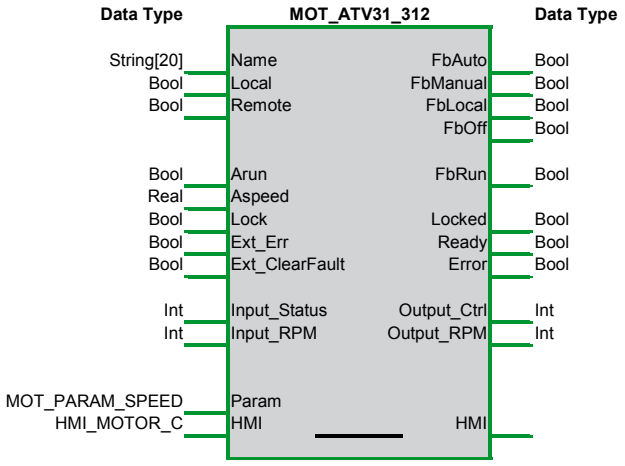
During the configuration of the ATV 32 in the FDT (Field Device Tool) Container (PRM configuration tool in Unity Pro), the GSD offers a choice among 4 telegrams. In our case, the telegram 102 is chosen, it allows for communication with 6 input and 6 output words. However, the DFB developed for the ATV 32 with Profibus DP does not use all the words (see table below).

The input and output words are automatically generated with Unity Pro FDT Container during the configuration phase. Refer to chapter 4.2.8 on Page 80 for more details.

Note: As this DFB controls a variable speed drive, an input pin named ASpeed has been added so the PAC can specify a speed setpoint for the drive when it runs in automatic mode.

	DFB Pin	Internal Register	Type	Description
Input	Input_Status	3201 MSB	BYTE	Status Register
		3201 LSB	BYTE	Status Register
		8604 MSB	BYTE	Actual speed
		8604 LSB	BYTE	Actual speed
	Input_Meas	3204	INT	Motor current
		3208	INT	Motor voltage
		Free	INT	-
		Free	INT	-
Output	Output_Ctrl	8501 MSB	BYTE	Control Register
		8501 LSB	BYTE	Control Register
		8602 MSB	BYTE	Speed setpoint
		8602 LSB	BYTE	Speed setpoint

5.2.8. ATV 31 / ATV 312 Variable Speed Drives on CANopen



The communication with these devices uses 2 input words and 2 output words. These words correspond to the variables automatically generated during the configuration performed with Advantys associated with Unity Pro. Refer to chapter 4.2.4 on Page 72 for more details.

	DFB Pin	CANopen register	Type	Description
Input	Input_Status	6041	INT	Status Register
	Input_RPM	6044	INT	Actual speed
Output	Output_Ctrl	6040	INT	Control Register
	Output_RPM	6042	INT	Speed setpoint

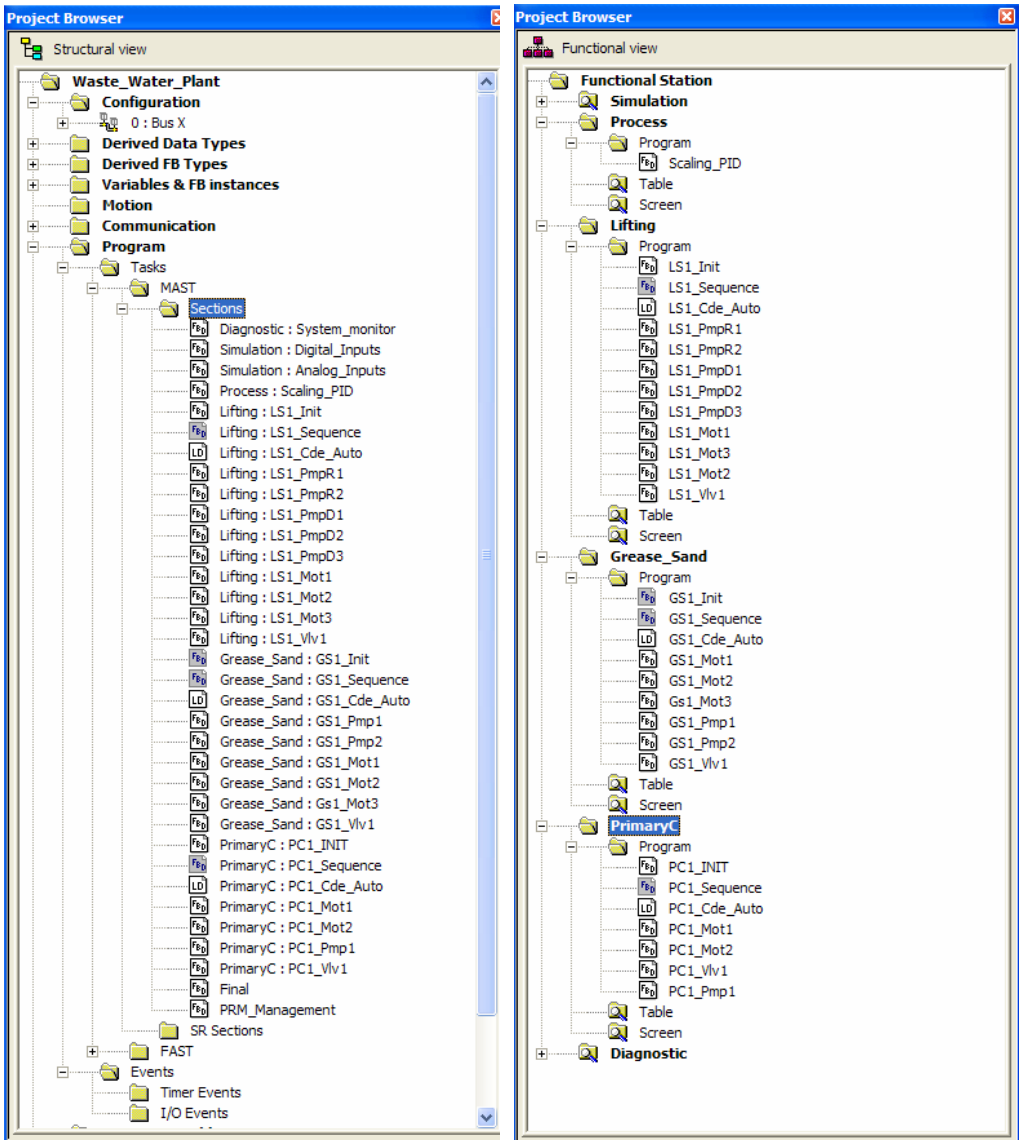
5.3. Unity Pro Program Structuring

5.3.1. Sections organization

The Unity Pro program consists of several sections.

In Unity Pro, the program can be represented in two ways:

- using a structural view, directly related to the PAC application
- using a functional view, allowing greater readability of the process. A good way to structure a program is by defining functional modules. This method has multiple advantages: the clarity of the program, useful for maintenance or development, and the ability to easily duplicate a process functional unit by export/import.



5.3.2. Variable Naming Rules

For the consistency of the project, the names of the variables that are to be exchanged with the SCADA system are defined before developing the application. In our project, we use the naming rule described below.

The name of the variable is composed of 3 parts: XXX_Yyy_Zzz

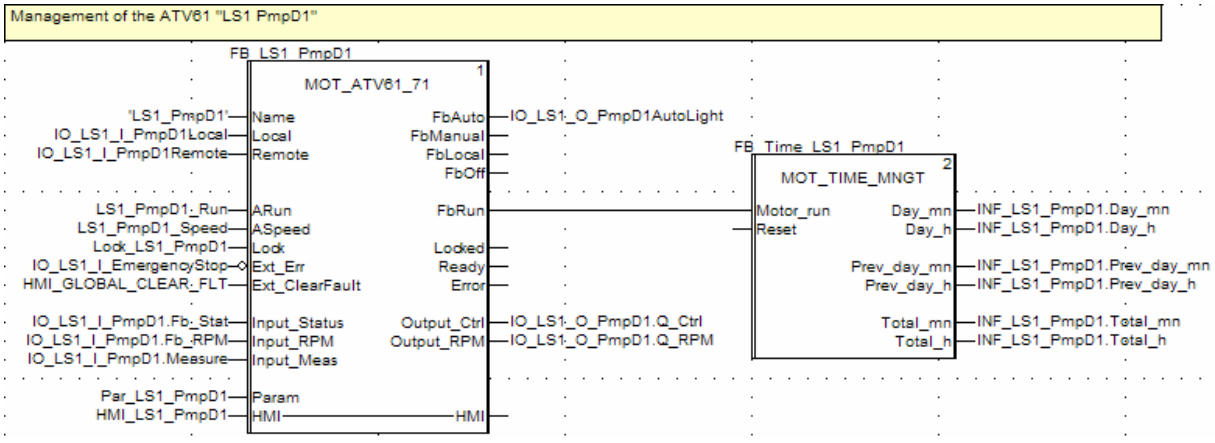
- XXX is the variable group

In our application, the possible values of the variable group are **HMI** or **INF**. These valus correspond to the variables exchanged with the SCADA system.

- Yyy is the identification of the functional unit to which the device belongs
- Zzz is the identification of the device type such as pump, motor, valve

5.3.3. DFB instantiation example

The picture below shows an example of a MOT_ATV61_71 DFB instantiated in Unity Pro.



All the DFBs are instantiated the same way. For more details, refer to the Unity Pro Water project provided with this guide.

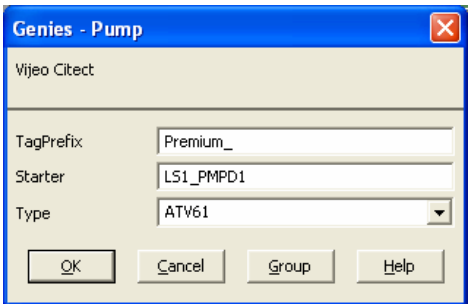
5.4. SCADA Implementation

5.4.1. SCADA Principle

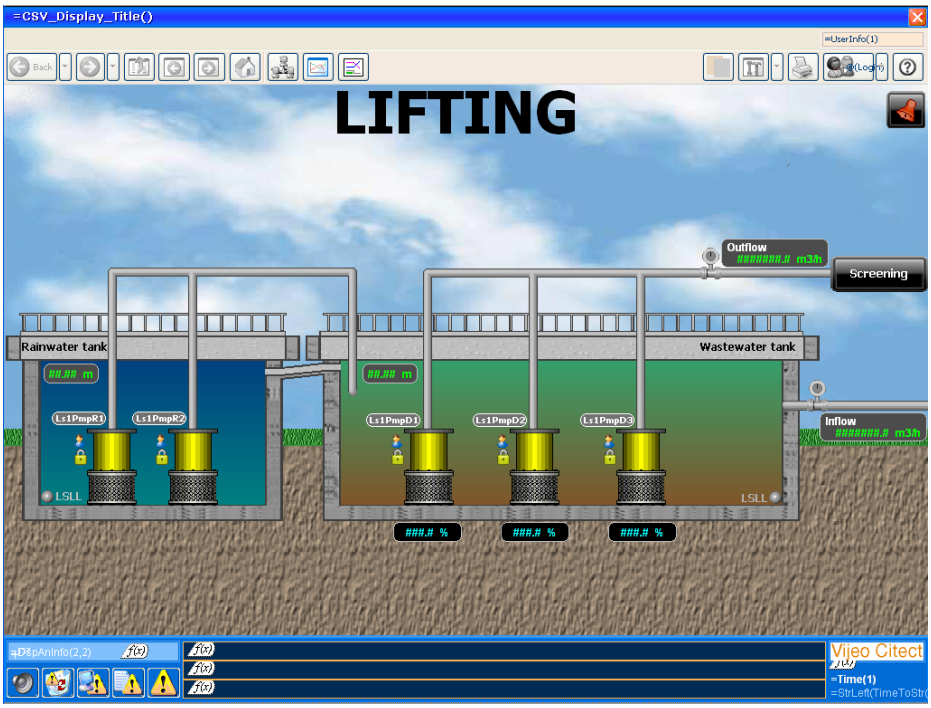
Genie and Super Genie objects are defined to control and monitor the motor control process. For each type of graphical object (pump, motor and so on) a Genie is created. This Genie can be copied from a Genie dialog box and added to the graphical page. When the Genie is pasted in the graphical page, a pop-up window is activated to substitute the tags used in the Genie and Super Genie. Each Super Genie is associated to a Cicode function. This function:

- substitutes tags in the Super Genie
- opens the associated Super Genie

For example, the screenshot below describes the instantiation of a Genie for the pump LS1_PMPD1 controlled by the ATV 61. The ATV 61 is piloted by the Premium I/O Device.



In the Water project, we have developed several pages for the representation of the whole process. As an example, the figure below presents the lifting unit page once every object has been instantiated.



6. Operation

6.1. Introduction

In this chapter, operations on process control, motor control diagnostic and FDR devices maintenance are described.

6.2. Process Control and Diagnosis

The system architecture developed for this guide includes various control and diagnostic solutions.

The following interfaces are used for process control and diagnostic functions:

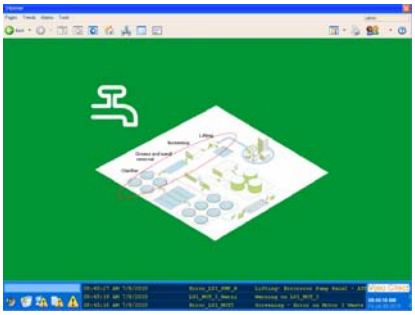
- The Vijeo Citect SCADA system, which provides complete process monitoring and the ability to control it in Manual mode.
- The web diagnostic services, which provide system diagnostics during the maintenance phases. Products such as PAC, ATV 61, Advantys STB and ETG100 gateway have an embedded web server with diagnostic pages that can be accessed from a standard web browser.
- Buttons and indicators on a local panel, which provide immediate display of the equipment status and permit local command operations on starters.
- Dedicated software tools such as Unity Pro or PowerSuite, which allow diagnosis of equipment and processes, particularly in the Design and Implementation phases of the process.

6.2.1. Vijeo Citect SCADA Interface

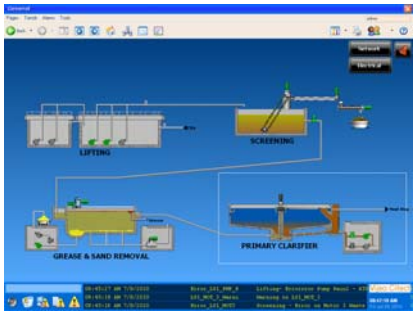
Pages

A number of pages have been developed for the monitoring and control of the pre-treatment part of a wastewater station.

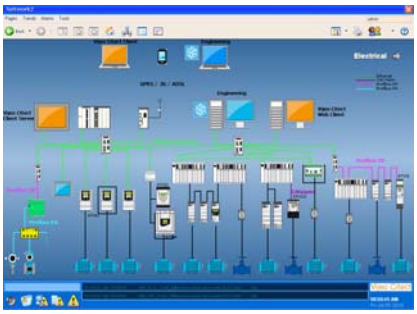
Home page:



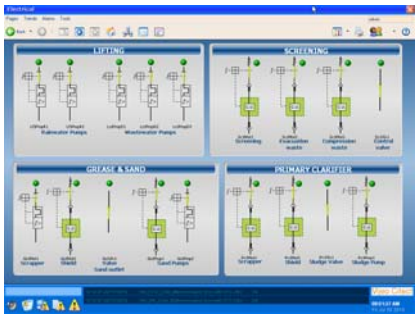
General page:



Network page:

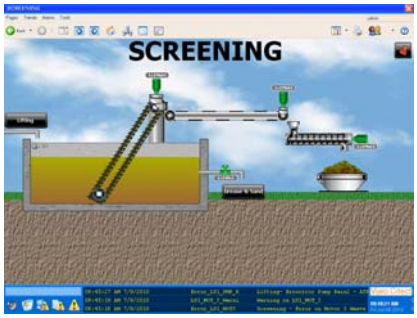


Electrical page:

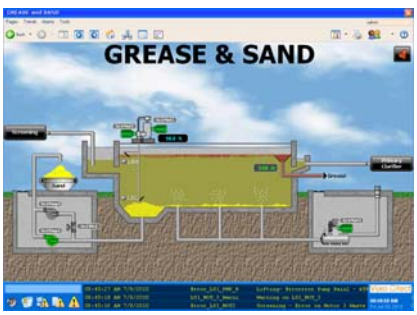


Lifting page:

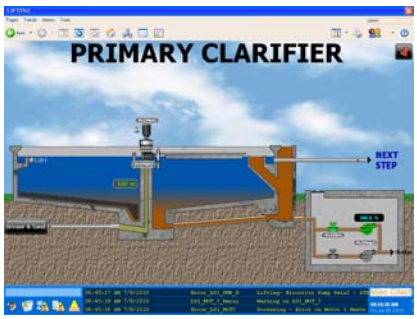
Screening page :



Grease & Sand Removal page:



Primary Clarifier page:

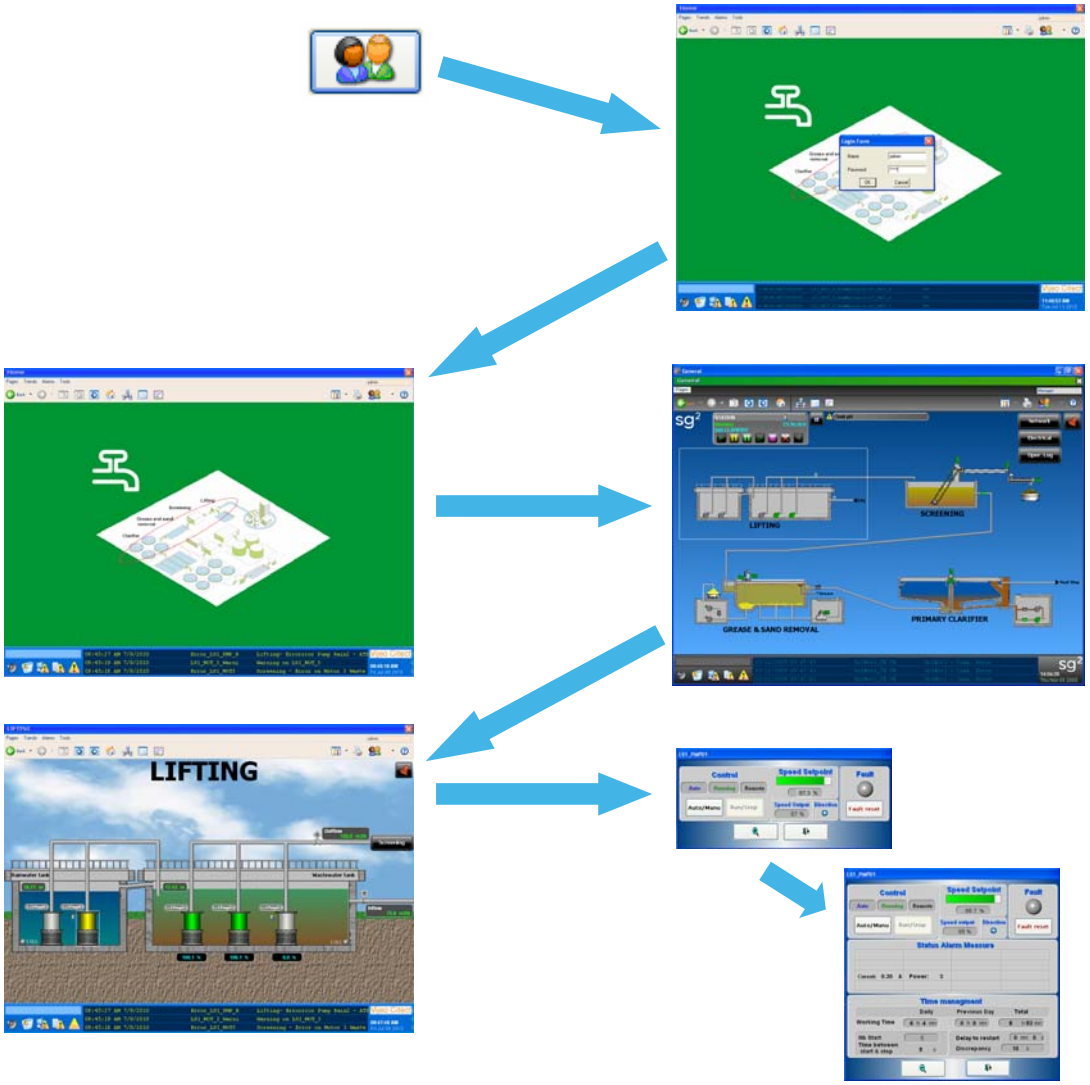


Navigation

The navigation takes place through a graphic environment. Once the Vijeo Citect project is launched, the system opens a home page, which situates the project in the whole water treatment plant. A simple click on the home page leads to the general view, which displays the project and its four pre-treatment units: Lifting, Screening, Grease & Sand Removal and Primary Clarifier. The mouse pointer highlights them within a square. Clicking on the desired unit displays the equipment included in the unit as well as each control module related to the equipment. The goal is to make the navigation intuitive: unit-> element-> equipment-> control module and so on.

A navigation toolbar is located at the top of the screen, and a page dedicated to alarms is located at the bottom. On the general page, the units with at least one alarm are surrounded by a blinking red box.

The following figures show the navigation sequence for accessing the control panel of the Lf1_PmpD1 pump in the Lifting unit.

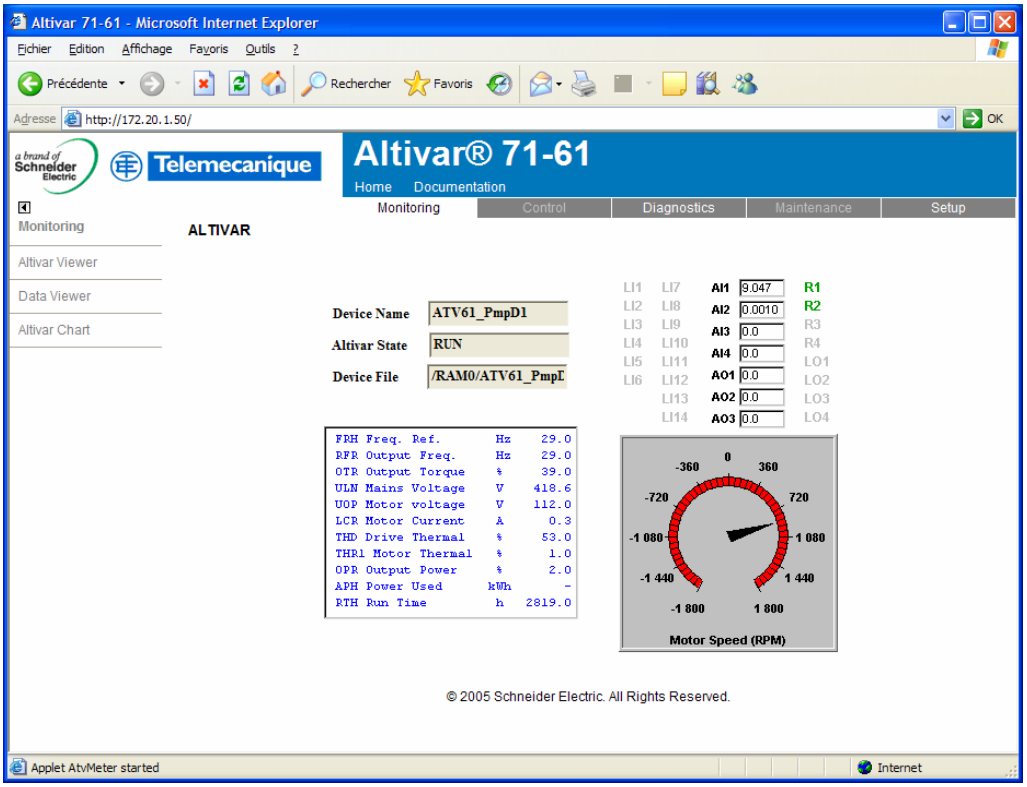


6.2.2. Web Interface

Some products have an embedded Web server. They only allow diagnostics with a Web browser.

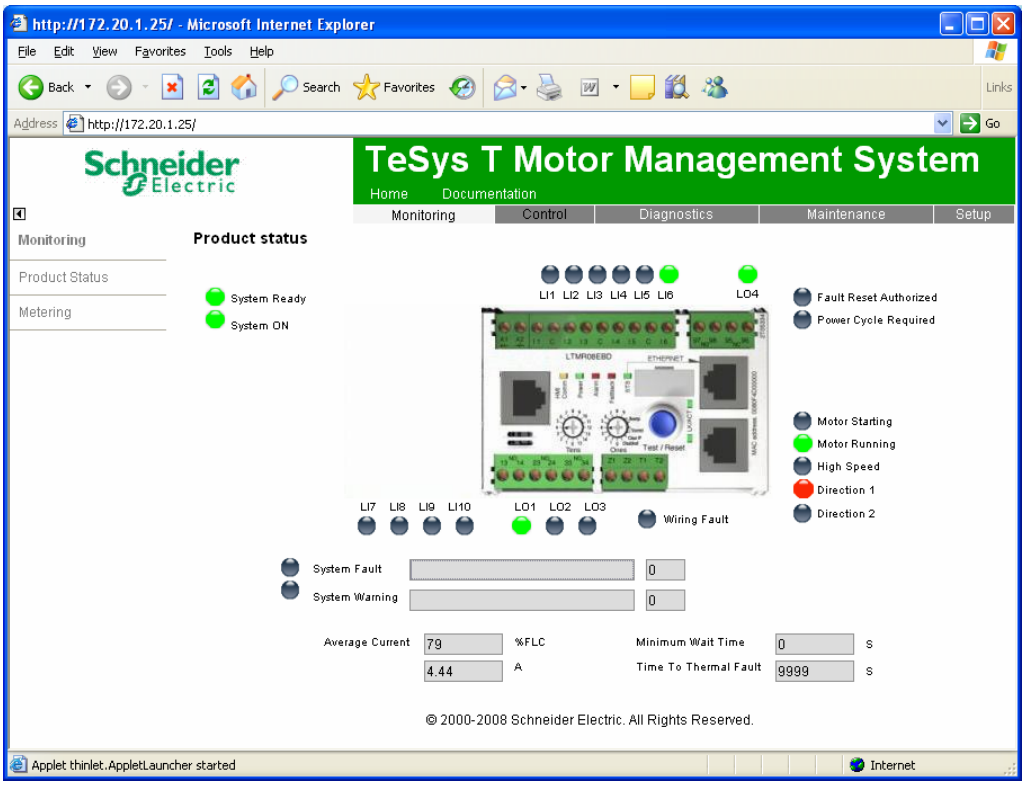
ATV 61 Variable Speed Drive of the Lf1_PmpD1 pump of the lifting unit:

In a Web browser, type the Lf1_PmpD1 ATV 61 drive IP address 172.20.1.50 in the address bar. Then, enter the default login “USER” and password “USER”.



TeSys T Controller of the Gs1_Mot3 motor of the Grease & Sand Removal unit:

In a Web browser, type the Gs1_Mot3 TeSys T controller IP address 172.20.1.25 in the address bar.

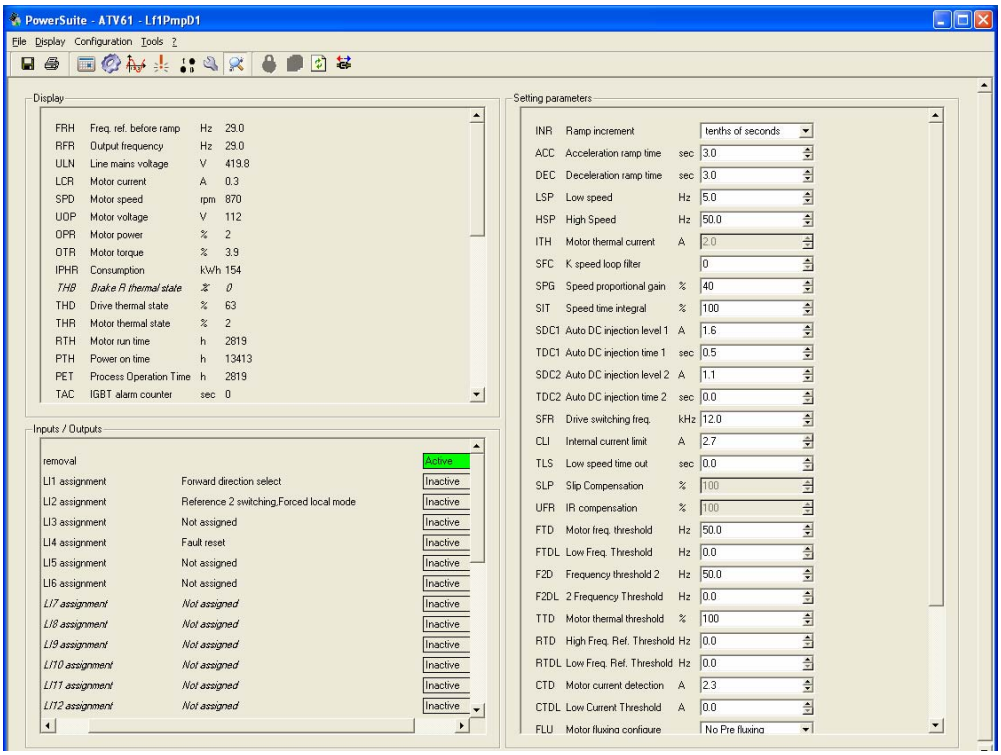


6.2.3. Software Tools

PowerSuite Diagnostic Tool

From a computer with the PowerSuite tool installed, you can access the different starters.

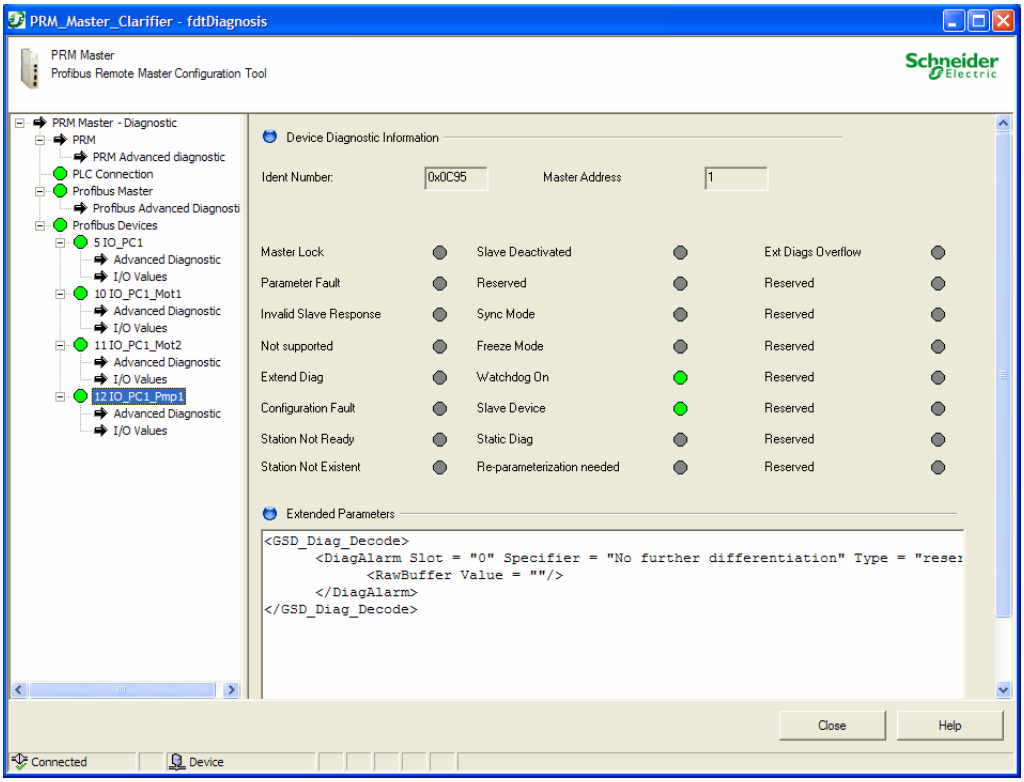
An example with the Lf1_PmpD1 ATV 61 drive of the lifting unit is shown below.


PowerSuite - ATV61 - Lf1PmpD1
File Display Configuration Tools ?
Display
FRH Freq. ref. before ramp Hz 29.0
RFR Output frequency Hz 29.0
ULN Line mains voltage V 419.8
LCR Motor current A 0.3
SPD Motor speed rpm 870
UOP Motor voltage V 112
OPR Motor power % 2
OTR Motor torque % 3.9
IPHR Consumption kWh 154
THB Brake R thermal state % 0
THD Drive thermal state % 63
THR Motor thermal state % 2
RTH Motor run time h 2819
PTH Power on time h 13413
PET Process Operation Time h 2819
TAC IGBT alarm counter sec 0
Inputs / Outputs
removal Active
LI1 assignment Forward direction select Inactive
LI2 assignment Reference 2 switching,Forced local mode Inactive
LI3 assignment Not assigned Inactive
LI4 assignment Fault reset Inactive
LI5 assignment Not assigned Inactive
LI6 assignment Not assigned Inactive
LI7 assignment Not assigned Inactive
LI8 assignment Not assigned Inactive
LI9 assignment Not assigned Inactive
LI10 assignment Not assigned Inactive
LI11 assignment Not assigned Inactive
LI12 assignment Not assigned Inactive
Setting parameters
INR Ramp increment tenths of seconds
ACC Acceleration ramp time sec 3.0
DEC Deceleration ramp time sec 3.0
LSP Low speed Hz 5.0
HSP High Speed Hz 50.0
ITH Motor thermal current A 2.0
SFC K speed loop filter 0
SPG Speed proportional gain % 40
SIT Speed time integral % 100
SDC1 Auto DC injection level 1 A 1.6
TDC1 Auto DC injection time 1 sec 0.5
SDC2 Auto DC injection level 2 A 1.1
TDC2 Auto DC injection time 2 sec 0.0
SFR Drive switching freq. kHz 12.0
CLI Internal current limit A 2.7
TLS Low speed time out sec 0.0
SLP Slip Compensation % 100
UFR IR compensation % 100
FTD Motor freq. threshold Hz 50.0
FTDL Low Freq. Threshold Hz 0.0
F2D Frequency threshold 2 Hz 50.0
F2DL 2 Frequency Threshold Hz 0.0
TTD Motor thermal threshold % 100
RTD High Freq. Ref. Threshold Hz 0.0
RTDL Low Freq. Ref. Threshold Hz 0.0
CTD Motor current detection A 2.3
CTDL Low Current Threshold A 0.0
FLU Motor fluxing configure No Pre fluxing

Unity Pro Diagnostic Tool

From Unity Pro, some devices propose an advanced diagnostic tool.

The ATV 32 behind PRM Master DTM can be diagnosed directly from Unity Pro using PRM Diagnostic, as described in the screenshot below.



6.2.4. Local Panel Interface

ATV 61 graphic display terminal provides a quick view of the equipment status and allows local command operations on the starter.

6.3. Faulty Device Replacement (FDR) Operation

This chapter details the maintenance procedures to replace the TeSys T of the Gs1_Mot3 motor of the Grease & Sand Removal unit and the ATV 61 of the pump Lf1_PmpD1 of the Lifting unit. The FDR service has previously been configured on these devices (see chapter 4.2.1 on Page 56).

6.3.1. TeSys T

The following table describes the steps to replace the TeSys T of the Gs1_Mot3 motor:

Step	Action
1	Power off the TeSys T to be replaced and unplug all the cables.
2	Install and wire the replacement TeSys T controller.
3	Use the rotary switches on the front of the replacement TeSys T controller to assign it the same Device name **TeSysT001** by setting the Tens switch on “0” and the Ones switch on “1”.
4	Power on the controller.

6.3.2. ATV 61

The following table describes the steps to replace the VSD of the pump Lf1_PmpD1:

Step	Action
1	Power off the ATV 61 to be replaced and unplug all the cables.
2	Install and wire the replacement ATV 61. Do not connect the Ethernet cable yet.
3	Power on the ATV 61.
4	Configure the ATV 61 Device name: From the menu **COMMUNICATION->ETHERNET**, select the sub-menu **DEVICE NAME** and type the same device name: **ATV61_PmpD1**. **Note:** IP parameters (IP Card, IP Mask, IP Gate and IP Master) must be set to 0.0.0.0 to allow the DHCP service to run.
5	Activate the FDR service. From the menu **COMMUNICATION->ETHERNET**, select the menu **FDR** validation and select **Yes** to activate the FDR service.

6	Turn the drive off and then back on again to have the device name taken into account.
7	Connect the drive to the Ethernet network.

7. Performance

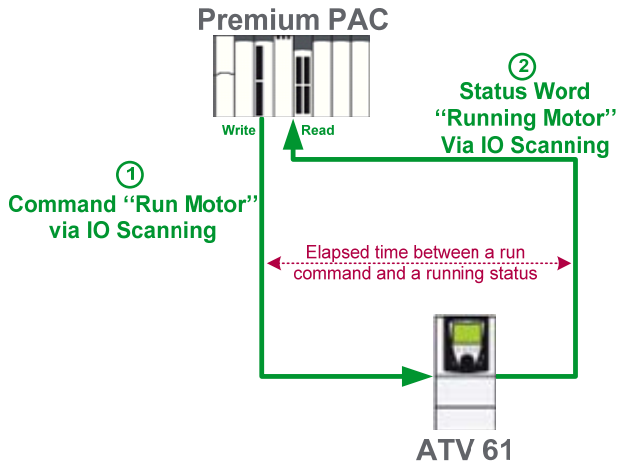
The following performances are measured using a 1 ms FAST task.

This chapter comprises the results for the following performance measurements:

- ART
- FDR

7.1. Application Response Time (ART) of Devices

The ART is determined by measuring the time delay from when the request for “Run Motor” command is issued by the IO Scanning to when the PAC receives “Running” status back from the starter.


Premium PAC
Write
Read
② Status Word “Running Motor” Via IO Scanning
① Command “Run Motor” via IO Scanning
Elapsed time between a run command and a running status
ATV 61

The following table summarizes the ART measurements performed on our architecture.

Devices on Ethernet network	MAST task period : 60 ms IOScanning repetetive rate : 30 ms	MAST task period : 120 ms IOScanning repetetive rate : 60 ms
ATV 61 Ethernet	110 ms	172 ms
TeSys T Ethernet	216 ms	268 ms
ATS 48 Modbus via ETG100	no suitable on Modbus serial line	713 ms
TeSys U on Advantys STB island via EPI2145	114 ms	210 ms
TeSys U on Advantys STB extension bus	115 ms	208 ms

Devices on Profibus DP network	MAST task period : 60 ms IOScanning repetetive rate : 10 ms	MAST task period : 120 ms IOScanning repetetive rate : 10 ms
TeSys U Profibus DP via PRM	202 ms	232 ms
ATV 32 Profibus DP via PRM	54 ms	112 ms

Profibus DP parameters: - Profibus BaudRate: 1500 kbit/s

- Profibus Cycle: 8ms

7.2. Faulty Device Replacement (FDR)

These results are the measurement of the time needed by the device to recover an operational state.

To measure this time accurately, the device is disconnected from the Ethernet network. While the device is offline, a configuration parameter is changed on the device. When the device is connected back on the Ethernet network, the FDR server sends the operational configuration to the device and sets all the parameters to their runtime values. The monitoring of the parameter that has been modified offline allows detecting when the device recovers an operational state (when the parameter turns back to its runtime value, the device is declared operational).

Thus, the recovery time is defined by the time needed by the device to get back to an operational state. The following table summarizes the recovery times measured in our architecture.

Device	MAST task period : 60 ms IOScanning repetetive rate : 30 ms
ATV 61 Ethernet	21 s
TeSys T Ethernet	24 s

Schneider Electric Industries SAS

Head Office
89, bd Franklin Roosvelt
92506 Rueil-Malmaison Cedex
FRANCE

www.schneider-electric.com

Due to evolution of standards and equipment, characteristics indicated in texts and images in this document are binding only after confirmation by our departments

Print:

Version 2.0 – 07 2010